- 第1章_绪论
- 第2章_空间现象抽象表达
- 第3章_空间数据模型
- 第4章_空间数据组织与管理
- 第5章_空间数据索引技术
- 第6章_空间数据查询、访问
- 第7章_时态空间数据库
- 第8章_空间数据元数据与空间数据共享
- 第9章_空间数据库设计

绪论

主讲老师：赵 娜
联系方式：zhaona@tjau.edu.cn

- 闭卷考试
- 成绩 = 平时成绩（40%）+ 期末成绩（60%）
  - 平时成绩
    - 出勤情况
    - 课堂考查

- 实验成绩

❖哪些是空间数据?

- 日常生活中最直观,最经常用到的空间数据
  - 地图——地理空间数据
- NASA卫星图像数据
  - 千兆字节每天
- 气候数据
- 医学图像数据

❖哪些是非空间数据?

- 姓名,电话号码,邮件地址...

❖地图

- 运用一定的数学法则与地图语言,经过制图综合,将地理信息表现在平面

上。

- 地理信息：描述地表形态及其所附的自然和人文地物特征和属性的总称。

❖地图制图自动化

- 随着计算机技术的发展
- 摆脱手工制图

❖地图数据库系统技术

- 地图数据
  - 地图比例尺影响——详细程度不同
  - 强调数据可视化——忽略了空间关系
  - 按地图印刷色彩分层管理——割裂了地物之间的联系
  - 地图图幅限制了数据范围

- 传统数据库管理系统——非空间数据
  - 允许并发连接数据
  - 具有稳定的搜索查询功能——大数据集
  - 具有高效的非空间数据查询功能,但不适用于空间数据
- 非空间数据查询举例
  - 列出拥有书的数目在1000以上的书店的名称
  - 列出在2001年刷卡消费的消费者姓名
- 空间数据查询举例
  - 列出距离天津市10公里以内的书店的名称
  - 列出居住在天津市或者其相邻城市的刷卡消费的消费者姓名

- 很多重要的领域需要对空间数据进行查询
  - 军队指挥官——敌人从昨晚起是否有明显的调动迹象？
  - 保险公司风险经理——黄河沿岸的哪些家庭最可能受到洪水的影响？
  - 医生——根据该病人的核磁共振图像，我们是否医治过类似病症的病人？
  - 气象学家——怎样才能测试和检验新研究出的全球变暖模型？

- 地理信息系统
  - Geographic Information Systems
  - 空间数据分析软件系统
    - 科学界和政府部门的专家
- 地理信息服务
  - Geographic Information Services
  - 提供地理或空间服务
    - Google earth

- 电子地图

❖ 空间（Space）
  - 物理学: 宇宙在3个互相垂直的方向上所具有的广延性。
  - 天文学: 时空连续体系的一部分。
  - 地理学: 在地球表层空间实体集上的关系。

❖ 地理空间（Geospace）
  - 该空间是一个相对空间,是一个空间实体组合排列集,强调宏观的空间分布和空间实体间的相关关系。
  - 数学描述

    设$E_1, E_2, E_3, \ldots, E_n$为n个不同类的地理空间实体;

    R表示地理空间实体值的相互联系,相互制约关系;

    $\Omega = \{E_1, E_2, E_3, \ldots, E_n\}$表示地理空间中各组成部分(实体)的集合;

则地理空间可以表示为 S={Ω,R}

- 空间数据（Spatial Data）
  - 数据的一种特殊类型,指凡是带有空间坐标的数据
    - 建筑设计图数据
    - 机械设计图数据
    - 各种地图数据

- 地理空间数据(Geospatial Data)
  - 空间数据的一种特殊类型,指带有地理坐标的数据
  - 地理实体的空间特征和属性特征的数字描述
    - 空间特征（定位）：几何特征和实体间空间关系

- 属性特征（定性）：数量特征，质量特征和时间特征

- 本课程研究主要研究地理空间数据，在以后的讲解中统称为空间数据。

❖ 空间数据的组成

- 空间实体
  - 对空间存在于自然世界中的地理实体的抽象，主要包括点、线、面以及实体等基本类型。
- 空间关系（拓扑关系）
  - 空间对象建立之后，进一步定义其相互之间的关系。

❖ 空间数据的复杂性

- 数据类型多——几何数据，关系数据
- 数据操作复杂——定位检索，拓扑关系检索等
- 数据输出多样——数据，报表，图形
- 数据量大——来源多样，测量数据，

图形图像数据等

- 空间数据库
  - 描述与特定空间位置有关的真实世界对象的数据集合
  - 既能处理空间参考对象类型,也能处理非空间参考对象类型

- 根据数据来源，概括为4种
  - 地图数据
  - 影像数据
  - 地形数据
  - 属性数据

- 卫星遥感影像

- 数字高程模型（Digital Elevation Model，缩写DEM）

- 数字高程模型是在某一投影平面(如高斯投影平面)上规则格网点的平面坐标（X，Y）及高程（Z）的数据集。
- 数字高程模型是描述地表起伏形态特征的空间数据模型,根据不同的高程精度，可分为不同类型。为完整反映地表形态,还可增加离散高程点数据。

❖根据表示对象

- 类型数据
- 面域数据
- 网络数据
- 样本数据
- 曲面数据
- 文本数据
- 符号数据

- 1. 时空特征
  - 空间数据一般由空间、属性和时间三部分构成
    - 空间特性指空间实体的空间位置及其与其他空间实体的空间关系
      - 绝对空间位置
      - 相对空间位置
      - 空间关系:如拓扑关系,顺序关系,度量关系
    - 属性特征指地学现象的数量、质量和分类等属性信息
    - 时态特性指地理数据采集或地理现象发生的时刻或时段

- 空间特征是空间数据最基本的特征
- GIS区别于其它的软件的根本特征
- 记录的是空间实体的位置、拓扑

关系和几何特征，是由于地物或现象的空间分布所产生

- ❖通过特定空间参照系下的坐标直接表达
  - 经纬度坐标
  - 标准的地图投影坐标
  - 任意的直接坐标等
- ❖基于坐标的派生数据
  - 定量的度量信息：面积、周长、质心、距离等
  - 定性的空间关系：拓扑关系、方位关系

- ❖描述地物所固有的，不是由于地物空间分布所带来的特征
  - 如某地的年降雨量、土地酸碱类型、人口密度、交通流量、空气污染程度等。
- ❖这类数据在其它类型的信息系统

中均可存储和处理

- ❖属性数据通常以数字、符号、文本和图像等形式来表示

- ❖指地理实体的时间变化或数据采集的时间等
- ❖空间数据涉及时间特征的几个方面
  - 地物的生命周期（产生、消亡）
  - 地物的移动（移动点）
  - 属性的时效性
- ❖相关的问题
  - 时间关系 → 时空拓扑关系

- ❖ 2. 多维特征

- 3. 多尺度性
  - 不同级别的系统在空间规模和时间长短方面存在很大差异，且由于空间认知水平、精度和比例尺等不同，地理实体的表现形式也不相同
    - 空间多尺度：空间范围大小或地球系统中各部分规模的大小，可分为不同的层次。
    - 时间多尺度：地学过程或地理特征有一定的自然节律性，且时间周期长短不一。
- 4. 海量数据特征

- 空间数据处理与更新
  - 数据的整体更新、局部更新、采集途径、时效性、保持原有数据的不变、更新数据与原有数据正确连接等
- 海量数据存储与管理
  - 空间数据库的布局和存取能力对地

理信息系统功能的实现和工作效率影响极大

❖空间分析与决策

❖空间信息交换与共享

1.1.2 空间数据库的发展历史

1.文件系统阶段

- 20世纪60年代早期
  - 数据管理方法的雏形
  - 特点
    - 文件在外存中的物理结构与用户观点的逻辑结构完全一致
    - 组织方式为顺序式
  - 缺点
    - 只能应付批处理，不适应于实时访问
    - 用户各自建立文件，不适宜共享，冗余度高
    - 物理结构发生变化或更换外存，就需要修改或重编应用程序

2.20世纪70年代

- 尝试将空间数据进行数据库管理的方法，将空间数据中的点、线、面分别存储和管理，点可以进行结构化管理，线和面用相邻两点进行结构化管理

3.20世纪80年代（文件-数据库混合管理）

- 1981年ESRI推出第一个商用地理信息产品Arc/Info
  - 对用户观点的数据进行严格细致的描述，使得文件、记录、数据项等单位之间的联系清晰，结构简单。
  - 允许用户以记录或数据项作单位进行访问，也允许多关键字检索和文件之间的交叉访问
  - 数据的应用独立于数据的存储
- 混合管理模式
  - 几何图形数据——文件系统
  - 属性数据——商用数据库管理系统
- 不是真正意义上的空间数据库管理系统

全关系型数据库管理系统（RDBMS）

- 20世纪70 年代后期
- 特点
  - 图形和属性数据都使用现有的关系数据库管理系统管理
  - 在现有数据库管理系统的基础上，用户自主开发，使之可以管理非结构化的图形数据
- 系统模型

4.20世纪80年代后期到90年代初期

- 要实现数据库一体化存储和管理(即空间数据与属性数据一起存入数据库）
- 主要难题：空间数据是变长的
- 解决方法：对象关系型数据库
  - 扩展通用的关系数据库管理系统，使之直接存储和管理非结构化的空间数据

- 缺点
  - 没有解决对象的嵌套问题
  - 空间数据结构不能由用户任意定义使用上受到限制

面向对象的数据库系统(OODBS)

- 采用面向对象方法建立的数据库系统
- 面向对象方法的基本思想
  - 对问题领域进行自然分割，以更接近人类通常思维的方式建立问题领域的模型，以便对客观的信息实体进行结构模拟和行为模拟，从而使设计出的系统尽可能直接地表现问题求解的过程。
- 特点
  - 适应空间数据的表达和管理
  - 支持变长记录，对象嵌套，信息继承和聚集
  - 面向对象的空间数据库管理系统允许用户定义对象和对象的数据结构以及他的操作
- 缺点

- 技术不成熟
- 价格昂贵

5.20世纪90年代中后期

- 1996年美国ESRI公司与Oracle公司合作，开发出空间数据库引擎（简称SDE）
- 其他研究热点
  - 空间数据仓库
  - 空间数据联机分析
  - 空间数据挖掘

❖空间数据管理模式现状

- 分析5种空间数据管理方式的优缺点

❖空间数据模型现状

❖空间数据库存在问题

- 数据共享
- 数据“瓶颈”问题

- 数据安全

❖ 空间数据库特征
- 综合抽象特征
- 非结构化特征
- 分类编码特征
- 复杂性与多样性

❖ 空间数据库与传统数据库的差异
- 信息描述差异
- 数据管理差异
- 数据操作差异
- 数据更新差异
- 服务应用差异

❖ 分布式数据库
❖ 专家数据库
❖ 演绎数据库
❖ 多媒体数据库
❖ 工程数据库

❖分布式数据库

- 使用较小的计算机系统,每台计算机可单独放在一个地方,每台计算机中都有DBMS的一份完整拷贝副本，并具有自己局部的数据库,位于不同地点的许多计算机通过网络互相连接,共同组成一个完整的、全局的大型数据库。

❖1. 分布式数据库的特色

- （1）地方自治性
- （2）相互协作性
- （3）位置透明性
- （4）副本的透明性

❖2. 使用分布式数据库的原因

- （1）组织和经济上的需要

- （2）如何充分利用已有的数据资源
- （3）新的功能和结构增长
- （4）通讯开销
- （5）小型计算机的发展
- （6）网络技术的商品化

❖ 3. 分布式数据库的体系结构所包含的基本部件
- （1）数据库管理DB系统，即集中式数据库管理系统DBMS;
- （2）数据通讯子DC系统;
- （3）全局数据字典DD（有关网上的数据分布）;
- （4）分布式数据库管理DDB系统，即负责分布处理的数据管理。

❖ 上述四部分合起来称为分布式数据库管理系统DDBMS(Distributed Database Management Systems)。

❖ DDBMS提供的典型功能
- （1）存取其他结点的数据;（2）分布透明性;（3）支持数据库管理和控制;（4）对分布事物的并发控制和恢复等。

- DDBMS一个重要的问题是系统是均质的还是异质的。即硬件、操作系统和DBMS是否相同
- 最重要的是DBMS是否相同，每个结点都采用相同的DBMS，这种系统称为均质系统，否则称为异质系统。

- 专家数据库是人工智能与数据库技术想结合的产物。它具有两种技术的优点，而避免了它们的缺点。
  - 人工智能是研究计算机模拟人的大脑和模拟人的活动的一门科学,因此逻辑推理和判断是其最主要的特长，但对于信息检索则效率很低。
  - 数据库技术是数据处理的最先进的

技术，对于信息检索有其独特的优势，但对于逻辑推理却无能为力。

- 1. 人工智能的弱点
  - （1）人工智能系统中的知识库中只含有少量的规则和事实。这是不能进入实用的原因之一。
  - （2）人工智能系统的效率极低，这是不能进入实用的原因之二。
- 2. 传统数据库系统的弱点
  - （1)不能进行逻辑推理和知识处理。
  - （2）不能管理复杂的类型对象，如CAD，CAM，RLSI，CASE等。
- 3. 专家数据库的研究目标
  - （1）专家数据库中不仅包含大量的事实，而且应包含大量的规则。
  - （2）专家数据库系统应具有较高的检索和推理效率，满足实时要求。

- （3）专家数据库应不仅检索，而且能推理。
- （4）专家数据库应能管理复杂的类型对象如CAD，CAM，CASE等。
- （5)专家数据库应能进行模糊检索。

❖4. 专家系统的研究成果

- （1）智能数据库接口
- （2）知识数据模型的发展
- （3）存储模型

❖演绎的含义是根据已知的事实和规则进行推理，回答用户提出的各种问题。演绎数据库也被称为逻辑数据库演绎关系数据库或虚关系数据库。换言之，它们具有很强的推理能力，这种推理能力起源于人工智能的研究。

- 演绎数据库、知识库与智能数据库
  - 共同之处是三者都是人工智能与数据库的结合，都是以数据库为基础，吸取了人工智能的成功技术的成果。
  - 不同之处
    - 数据库与知识库是不同的概念，前者管理数据，后者管理知识。
    - 演绎数据库与智能数据库均属于数据库范围，它们均以数据库为基础，吸取了人工智能的技术。与知识库不同。
    - 演绎数据库虽然也含有规则，但它含有的规则较少，而含有的数据却是大量的，这是与知识数据库不同的。
    - 智能数据库不仅应用人工智能中的逻辑推理思想，而且还应用人工智能中自然语言理解、语言识别，图象、文字处理等多种方法与技术于数据库，以求得更多的功能、性能的改善与提高。因此，从某种意义讲，演绎数据库是智能数据库的一部分

- 能存储和管理多种媒体的数据库称为多媒体数据库。

- 多媒体DBMS的体系结构
  - （1）在传统DBMS基础上实现
  - （2）多个DBMS协调方案
  - （3）多媒体DBMS方案
  - （4）存储核心层+应用层多媒体DBMS体系

- 工程数据库是指在工程设计中，主要是CAD/CAM中所用到的数据库。
- 由于在工程中的环境、要求不同，工程数据库与传统的信息管理中用到的数据库有着很大的区别。

- 1. 工程数据库的应用环境

- 在工程设计中有着大量的数据和信息要保存和处理。
  - 例如零件的设计模型、图纸上的各种数据、材料、工差、精度、版本等各种信息需要保存、管理和检索。

❖2. 工程数据库的特色

- （1）设计者是一个临时用户;
- （2)主要数据库是图形和图象数据;
- （3）数据库规模庞大;
- （4）设计处理的状态是直观的和暂时的;
- （5）设计的多次版本信息都要予以保存;
- （6）事务是长寿的，从设计到生产周期较长;
- （7）数据要求有序性;
- （8）数据项可多达几百项。

- 由于数据获取技术、网络技术和计算机技术的发展导致数据的几何数增长
- 数据处理方法匮乏
  - 以遥感数据为例
    - 每年采集的数据量以TB计（据不完全统计，SPOT有250TB；ESA有400TB...），而被应用的数据仅占获取数据的10-15%

- 空间数据挖掘的特点
  - 与传统数据挖掘的不同:
  - (1) 传统数据挖掘处理的是数字和类别,而空间数据则是一些更为复杂

的数据类型，例如：点、线、多边形等对象;

- (2) 传统数据挖掘通常具有显式的输入,而空间数据挖掘的输入则常常是隐式的;
- (3) 在传统数据挖掘中，有一个至关重要的前提假设:数据样品是独立生成的。而这一假设在空间分析中是不成立的。事实上，空间数据之间是高度自关联的。

❖空间数据挖掘的发展现状

- 国际上最著名且有代表性的通用SDM系统有：GeoMiner,Descartes和ArcViewGIS的S_PLUS接口。
  - 以上SDM系统的共同优点是把传统DM与地图可视化结合起来，提供聚类、分类等多种挖掘模式，但它们在空间数据的操作上实现方式不尽相同。
    - Descartes是专门的空间数据可视

化工具，它和DM工具Kepler两者联合在一起才能完成SDM任务。

- GeoMiner是在MapInfo平台上进行二次开发而成，系统庞大，造成较大的资源浪费。
- S_PLUS的局限在于，它是用一种解释性语言（Script），功能的实现比用C和C++直接实现要慢得多，所以只能使用与非常小的数据库应用。

空间现象抽象表达

主讲：赵 娜

zhaona@tjau.edu.cn

2.1 现实世界的认知
2.2 从现实世界到模型世界
2.3 空间实体描述
2.4 空间实体矢量表达
2.5 空间实体栅格表达

- 认知科学
  - 研究人类感知和思维信息处理的科学
- 空间认知
  - 对现实世界的空间属性包括位置、大小、距离、方向、形状、模式、运动和物体内部关系的认知，是通过获取、处理、存储、传递和解译空间信息，来获取空间知识的过程。
  - 研究内容: 信息的组织、地图符号的感知响应、地图的视觉比较
- 空间类型表现形式
  - 感知空间
  - 认知空间
  - 符号空间
- 空间认知模式
  - 空间特征感知

- 空间对象认知
- 空间格局认知

❖地理空间认知

❖空间认知的三层模型

- 外模式
- 空间概念数据模型
  - 基于对象模型
  - 场模型
  - 网络模型
- 空间逻辑数据模型
  - 层次模型
  - 网络模型
  - 关系模型
- 物理数据模型

- 由于职业、专业等的不同，人们所关心的问题、研究对象、期望的结果等方面存在着差异，因而对现实世界的描述和抽象也是不同的，形成了不同的用户视图，称之为外模式。

- **GIS空间数据模型由概念数据模型、逻辑数据模型和物理数据模型三个有机联系的层次组成。**
  - 概念数据模型是关于实体及实体间联系的抽象概念集
  - 逻辑数据模型是表达概念数据模型中数据实体（或记录）及其间关系
  - 物理数据模型是描述数据在计算机中的物理组织、存储路径和数据库结构

- 在GIS中与空间信息有关的信息模型有三个:
  - 场模型：强调空间要素的连续性，用于表示在二维或者三维空间中被看作是连续变化的数据。--栅格数据模型
    - 场模型表示了在二维或者三维空间中被看作是连续变化的数据。
  - 基于对象（要素）的模型：强调空间要素的离散性，可以详细地描述离散对象。 --矢量数据模型
    - 基于对象（要素）的模型强调了离散对象，根据它们的边界线以及组成它们或者与它们相关的其它对象，可以详细地描述离散对象。
  - 网络模型：强调空间要素的交互，如水系、交通网络。
    - 网络模型表示了特殊对象之间的交

互，如水或者交通流。

- OpenGIS
  - Open Geodata Interoperation Specification,OGIS-开放的地理数据互操作规范
  - 由美国OGC(OpenGIS协会，OpenGIS Consortium)提出。
- OGC是一个非赢利性组织
  - 目的是促进采用新的技术和商业方式来提高地理信息处理的互操作性(Interoperability)，OGC会员主要包括GIS相关的计算机硬件和软件制造商（包括ESRI, Intergraph,MapInfo等知名GIS软件开发商），数据生产商以及一些高等院校，政府部门等，其技术委员会负责具体标准的制定工作。
- OpenGIS的目标

- 制定一个规范,使得应用系统开发者可以在单一的环境和单一的工作流中,使用分布于网上的任何地理数据和地理处理。
- 它致力于消除地理信息应用(如地理信息系统，遥感，土地信息系统，自动制图/设施管理(AM/FM)系统）之间以及地理应用与其它信息技术应用之间的藩篱，建立一个无“边界”的、分布的、基于构件的地理数据互操作环境,与传统的地理信息处理技术相比，基于该规范的GIS软件将具有很好的可扩展性、可升级性、可移植性、开放性、互操作性和易用性。

❖什么是空间实体？

- 具有确定的位置和形态特征并具有地理意义的地理空间的物体。
  - 确定的位置和形态特征：至少在给

定的时刻，空间实体具有确定的形态。

- “确定的形态”不意味着空间实体必须是可见的、可触及的实体，也可以是不可见的。
- 地理意义：在特定的地学应用环境中，被确认为有分析的必要。

- 举例
  - 河流、道路、城市——可见
  - 境界、航线——不可见

❖空间实体的基本类型

- 点
  - 延展度为0的空间实体
- 线
  - 延展度为1的空间实体，有一定范围的点元素集合，表示相同专题点的连续轨迹
- 面
  - 延展度为2的空间实体，表示平面区

域大范围连续分布的特征

- 体
  - 延展度为3的空间实体，3D空间中有界面的基本几何元素

❖ 基于对象的空间模型

- 将研究的整个地理空间看成一个域，地理实体和现象作为独立对象分布在该空域中。

- 强调个体现象
- 主要描述不连续的，或具有确定边界的地理现象。
  - 与土地和财产的拥有者记录有关的应用
  - 江河，建筑物，管理区域

❖ 对象的特点

- 1. 被识别
- 2. 重要（与问题相关）
- 3. 可被描述（有特征）

❖基于场的空间模型

- 把地理空间的事物和现象作为连续的变量来看待，任意指定的空间位置都对应一个唯一的属性值
- 适用于具有一定空间内连续分布特点的现象
  - 地表的温度、土壤的湿度

- 根据应用不同，场可以表现为二维或三维
  - 例：空气污染物在空间中的分布——三维

❖场模型的主要类型

❖场模型

- 根据属性的表示方法
  - 空间框架
  - 场函数
  - 场操作
    - 局部操作
    - 聚焦操作
    - 区域操作

❖场的特征

- 空间结构特征和属性域
  - 空间结构：规则的或不规则的，分辨率和位置误差十分重要。
  - 属性域：数值包括名称、序数、间隔和比率；支持空值。
- 连续的、可微的、离散的

- 各向同性和各向异性
  - 空间场内部的各种性质是否随方向的变化而发生变化
  - 空间场可以分为各向同性场和各向异性场
- 空间自相关
  - 空间场中数值聚集程度的

度量。

- 描述某一位置上的属性值与相邻位置上的属性值之间的关系

❖ 不互相排斥，可以共存
  - 在GIS数据结构设计和应用上经常采用两者集合
  - 例如
    - 降雨区域范围在特征上是场模型
    - 降雨量数据的采集点在空间上是分散的，无规律的对象模型
❖ 不同的思维方式
❖ 不同的数据表示方法
  - 对象模型——矢量数据结构
  - 场模型——栅格数据结构

❖ 矢量方法强调了离散现象的存在，由边界线（点、线、面）来

确定边界，因此可以看成是基于要素的。

- 矢量数据模型将现象看作原形实体的集合，且组成空间实体。在二维模型内，原型实体是点、线和面；而在三维中，原型也包括表面和体。
  - 矢量数据模型的表达
    - 通过记录坐标的方式尽可能精确地表示点、线、面等地理实体。
      - 点：由一对地理坐标定义
      - 线：用一连串有序的两个或多个坐标对集合表达对于本身宽度在研究中可以忽略的线状空间对象。
      - 面：通过对边界线的定义来进行

- **定位明显**
  - 其定位是根据坐标直接存储的，无需任何推算
- **属性隐含**
  - 属性则一般存于文件头或数据结构中某些特定的位置上
- 优缺点
  - 优点
    - 方便的表达空间实体之间的拓扑空间关系，图形精度高、数据存储量小、容易定义和操作单个目标、方便实现坐标变换和距离计算等
  - 缺点
    - 缺乏与遥感及数字地面模型直接结合的能力，难于处理叠置操作

- 利用欧几里得几何学中的点、线、面及其组合体来表示地理实体空

间分布的一种数据组织方式。

- Spaghetti（面条）结构
- 拓扑结构

❖空间数据按照基本的空间对象为单元进行单独组织，用一系列坐标串表示地物，记录空间实体的形状信息，不考虑空间实体间的关联关系等拓扑关系。

❖拓扑关系需要经过大量的计算得出

❖适用于制图系统

❖Spaghetti结构

- 面向多边形组织数据,将多边形看作是线的简单闭合,不属于任何多边形的线和点，另外组织。
- 采用坐标序列法进行编码

❖主要特点

- 结构简单，编码容易，直观，便于屏幕显示
- 缺点：数据冗余，可能导致公共边出现微小间隙或重叠；缺乏多边形邻域信息，和图形拓扑关系；
- 一般在计算机制图中使用；

❖ESRI Shapefile（shp）

- 美国ESRI公司开发的空间数据开放

格式。该文件格式已经成为了地理信息软件界的一个开放标准

- ESRI shapefile基本文件
  - *.shp: 主文件（图形文件）
    - 保存元素的几何实体 ，坐标文件
  - *.dbf: DBASE表（属性文件）
    - 保存关于元素的属性信息
  - *.shx: 索引文件
    - 保存几何实体索引

- Shp
  - 主文件是一个直接存取,变长度记录的文件,其中每个记录描述构成一个地理实体（Feature）的所有vertices坐标值。
- 索引文件
  - 每条记录包含对应主文件记录距离主文件头开始的偏移量
- dBASE表包含SHP文件中每一个Feature的特征属性，表中几何记

录和属性数据之间的一一对应关系是基于记录数目的ID。

- 在dBASE文件中的属性记录必须和主文件中的记录顺序是相同的。

❖ 图形数据和属性数据通过索引号建立一一对应的关系。

❖ 0 Null Shape
❖ 1 Point
❖ 3 PolyLine
❖ 5 Polygon
❖ 8 MultiPoint
❖ 11 PointZ
❖ 13 PolyLineZ
❖ 15 PolygonZ
❖ 18 MultiPointZ
❖ 21 PointM

- 23 PolyLineM
- 25 PolygonM
- 28 MultiPointM
- 31 MultiPatch

- 变长度空间数据
  - 有多条记录组成
- 记录组成
  - 固定长度的记录头
  - 变长度记录内容（空间坐标对）

- 记录头的内容
  - 记录号（Record Number），从1开始
  - 坐标记录长度（Content Length），16位字
- 记录内容
  - 目标的几何类型（ShapeType）
  - 具体的坐标记录（X，Y），主要包括

空Shape记录，点记录，线记录和多边形记录。

❖点记录

```
Point
   {
      Double X
      Double Y
    }
```

❖线记录

```
Line
{
   Double[4]  Box
   Integer
```

```
    NumPans
      Integer
    NumPoints
      Integer[NumPans]
    Parts
      Integer[NumPoints]
    Points
  }
```

❖面记录

```
    Polygon
    {
        Double[4]       Box
            Integer
    NumPans
        Integer
    NumPoints
```

```
        Integer[NumPans]
    Parts
        Integer[NumPoints]
    Points
        }
```

- 从抽象概念来理解其中图形元素间的相互关系，不考虑结点和线段坐标，只注意它们的相邻和连接关系
- 在拓扑模型中，空间目标的拓扑属性是在Spaghetti模型空间坐标基础上定义的，在地理信息系统中，拓扑生成意味着给Spaghetti文件增加拓扑结构
- 在拓扑模型中，多边形的边界被分割成一系列的弧和结点。弧、

结点和多边形之间空间关系在属性表中定义

- 弧段是最基本的空间数据单元之一，每个弧段包含两个结点——起结点和终结点，起结点和终结点定义了弧段的方向，从而也定义了该弧段的左右多边形；在结点之间有零个或多个拐点，弧段的长度和形状由结点和拐点的坐标所决定；
- 多边形由一系列的相互连结的弧段组成，并通过其内部的唯一标识点来标识。标识点的标识码和该多边形属性表中的标识码相一致，由此建立的多边形空间信息和属性信息的关系。
- 结点(Node)定义为弧段的起点、终点或几条线的交点。结点和拐点的差别在于结点具有拓扑特征，用于表示弧段是否

相连，而拐点没有拓扑特征，只是表示了弧段的位置和形状属性。

- 最基本的拓扑关系
  - 关联
  - 邻接
  - 包含

- 拓扑关系的表示
  - 结点拓扑关系
  - 线拓扑关系
  - 多边形拓扑关系

- 主要特点
  - 消除相邻多边形之间的重复线

- 拓扑信息与空间坐标分开存储
- 拓扑表必须在一开始就创建,需要花费一定的时间和空间
- 一些简单操作处理比较慢

❖ 优点

描述点、线、面的空间关系不完全依赖于具体的坐标位置。

空间关系信息丰富、简洁，数据冗余小。

方便多边形和多边形的叠合。

便于检查数据输入过程中的错误。

❖ 缺点

拓扑关系建立过程比较复杂

数据结构本身复杂

❖ 栅格数据模型

- 把空间看作像元(pixel)的划分，每

个像元都与分类或者标识所包含的现象的一个记录有关。

- 像元具有固定的尺寸和位置
- 像元的大小影响精度

❖特点

- 每个栅格中的像元位置被预先确定
  - 方便重叠运算

❖栅格数据结构

- 栅格数据结构实质
  - 规则像元阵列表示空间地物或现象分布的数据组织，用每个像元的行列号确定位置
  - 用每个像元的值表示地物或现象的非几何属性特征。
- 栅格结构表示的地表是不连续的，是量化和近似离散的数据。

- 每一个单元格对应一个相应的地块。

❖ 点实体：表示为一个像元；
❖ 线实体：表示为在一定方向上连接成串的相邻像元的集合；
❖ 面实体：表示为聚集在一起的相邻像元的集合。

❖ 栅格数据单元格经常是矩形（主要是正方形）的，但并不是必须如此。其单元格形状可以随应用的需要进行具体设定，比如设置为三角形。
❖ 栅格数据的比例尺就是栅格大小

与地表相应单元大小之比。

- 栅格尺寸越小，其分辨率越高，数据量也越大。

- 由于栅格结构对地表的离散，在计算面积、长度、距离、形状等空间指标时，若栅格尺寸较大，则造成较大的误差 。
- 由于栅格单元中存在多种地物，而数据中常常只记录一个属性值，这会导致属性误差。
  - 注意：遥感数据中的“混合像元”问题。

栅格数据：*坐标系与描述参数*

- 采用栅格模型的GIS，通常应用分

层的方法。在每个图层中栅格像元记录了特殊的现象的存在。每个像元的值表明了在已知类中现象的分类情况。

- 栅格模型的一个重要特征就是每个栅格中的像元的位置是预先确定的，描述同一区域的不同现象的栅格数据之间很容易进行逻辑运算。

- 栅格数据的特点
  - 属性明显
    - 数据中直接记录了数据属性或指向数据属性的指针，因而我们可以直接得到地物的属性代码
  - 定位隐含
    - 所在位置则根据行列号转换为相应的坐标，也就是说定位是根

据数据在数据集中的位置得到的。

- 栅格结构是按一定的规则排列的，所表示的实体的位置很容易隐含在网格文件的存储结构中

❖ 栅格数据的优缺点

- 优点
  - 结构容易实现，算法简单，易于扩充，修改，直观
  - 易于同遥感影像结合处理
- 缺点
  - 像元大小限制精度
    - 属性方面的偏差
    - 形态方面的畸形

❖ 原则

- 尽量保持地表的真实性，保证最大的

信息容量。

- ❖注意
  - 每一个单元可能对应多个地物种类或多个属性值。比如遥感图像中的“混合像元”。
- ❖常用方法
  - 中心点法
  - 面积占优法
  - 重要性法
  - 百分比法

- ❖中心点法
  - 处理方法
    - 用处于栅格中心处的地物类型或现象特性决定栅格代码
  - 常用于具有连续分布特性的地理要素
    - 如降雨量分布、人口密度图等。

- 面积占优法
  - 处理方法
    - 以占栅格区域面积比例最大的地物类型或现象特性决定栅格单元的代码
  - 面积占优法常用于分类较细，地物类别斑块较小的情况

- 重要性法
  - 处理方法
    - 根据栅格内不同地物的重要性，选取最重要的地物类型决定相应的栅格单元代码
  - 常用于具有特殊意义而面积较小的地理要素，特别是点、线状地理要素
    - 如城镇、交通枢纽、交通线、河流水系等，在栅格中代码应尽量表示这些重要地物

- 百分比法
  - 处理方法
    - 根据栅格区域内各地理要素所占面积的百分比数确定栅格单元的代码
  - 适用于地物面积具有重要意义的分类体系

- 栅格数据编码方法分为两大类
  - 直接栅格编码
  - 压缩编码方法
    - 链码
    - 游程长度编码
    - 块码
    - 四叉树

- 将栅格数据看作一个数据矩阵，逐行(或逐列)记录代码，可以每行都从左到右记录，也可以奇数行从左到右，偶数行从右到左。
- 这种记录栅格数据的文件常称为栅格文件，且常在文件头中存有该栅格数据的长和宽，即行数和列数。这样，具体的像元值就可连续存储了。
- 特点：处理方便，但没有压缩

❖ 压缩编码的目的就是用尽可能少的数据量记录尽可能多的信息，其类型分为

- 信息无损编码
  - 编码过程中没有任何信息损失，通过解码操作可以完全恢复原来的信息

- 信息有损编码
  - 为了提高编码效率，最大限度地压缩数据，在压缩过程中损失一部分相对不太重要的信息，解码时这部分难以恢复
- 在地理信息系统中的压缩编码多采用信息无损编码，而对原始遥感影像进行压缩时也可以采取有损压缩编码方法。

❖ 链码

- 又称Freeman编码或边界编码
  - 该编码方法将数据表示为由某一原点开始并按某些基本方向确定的单位矢量链。
- 主要记录线状地物或面状地物的边界
  - 优点：很强的数据压缩能力，并具有运算能力
  - 缺点：叠置运算比较难实施

- 游程长度编码
  - 把具有相同属性值的邻近栅格单元合并在一起，合并一次称为一个游程。游程用一对数字表达，其中，第一个值表示游程长度，第二个值表示游程属性值。每一个新行都以一个新的游程开始
- 实现方法有两种
  - 1. 只在各行（或列）数据的代码发生变化时依次记录该代码以及相同的代码重复的个数，从而实现数据的压缩。
  - 2. 逐个记录各行（或列）代码发生变化的位置和相应代码

- 第一种编码方案举例

- 第二种编码方案举例

- 游程长度编码优缺点
  - 优点

- 压缩效率较高，且易于进行检索，叠加合并等操作，运算简单，适用于机器存储容量小，数据需大量压缩，而又要避免复杂的编码解码运算增加处理和操作时间的情况

- 缺点
  - 对于图斑破碎，属性和边界多变的数据压缩效率较低，甚至压缩后的数据量比原始数据还大。

❖块码

- 将游程长度编码扩展到二维的情况
- 采用方形区域作为记录单元，每个记录单元包括相邻的若干栅格
- 数据结构
  - 初始位置（行、列号），半径，记录单位代码
- 优点
  - 具有可变分辨率。随图形复杂程度的提高而效率降低

- 在合并、插入、检查延伸性、计算面积等操作具有优越性

❖ 四叉树编码

- 常规四叉树的基本思想
  - 首先把一幅图像或一幅栅格地图等分成四部分，如果检查到某个子区的所有格网都含有相同的值（灰度或属性值），那么，这个子区域就不再往下分割；否则，把这个区域再分割成四个子区域，这样递归地分割，直至每个子块都只含有相同的灰度或属性值为止
- 采用四叉树编码时，为了保证四叉树分解能不断地进行下去，要求图像必须为$2^n \times 2^n$的栅格阵列，对于非标准尺寸的图像需首先通过增加背景的方法将图像扩充为$2^n \times 2^n$的图像。

- 运算量较大。
- 占用的存储空间较大。每个结点需要六个变量才能加以表达：一个变量表示父结点指针，四个变量代表四个子结点指针，一个变量代表本结点的灰度或属性值

❖ 四叉树的特点
- 具有可变的分辨率，有区域性质，压缩数据灵活
- 许多运算可以在编码数据上直接实现，提高了运算效率

❖ 规则四叉树
❖ 线性四叉树

- 一对四式四叉树

- 用五个字段来表示树中的每个结点，其中一个用来描述该结点的特性（有目标、空白、非结点），其余四段用于存放四结点的指针。
- 缺点
  - 大量的空间为指针所占用
- 优点
  - 方式自然，容易被人接受

- 将四叉树转换成一个线性表，表的每个元素与一个结点相对应，结点之间的层次关系在元素中描述。
- R1’ ab” ABCDc” EFGHd” IJKL2’

e”MNOPfg”QRSTh34’i”UVWXjkl

- 优缺点

  节省存储空间，对某些运算也是方便的。但是为此付出的代价是丧失了一定的灵活性

- 用五个字段表示每个结点，其中四个字段分别描述其四个子结点的状态（有目标，空白、非叶结点）；一个字段存放其子结点记录的地址（指针）。这里要求四个结点对应的记录是依次连续存放的。

- 紧凑的一对四式四叉树
  - 在存取某结点之前，需检查其它结点记录，看其有几个叶结点，以确定所

需结点记录。这种方法的存储需求无疑是最小的，但要有附加的计算量。

- 在原记录中增加一个字节一分为四（每个为2位）代表它的结点在指针指向区域中的偏移（见图3-2-10）。在找它的结点时，只要固定地把指针指向的位置加上这个偏移量（0¯3），就是所要的记录位置。这种方法适当地减少了计算量。

- 还有很多编码方法，如傅立叶变换、小波变换、余弦变换等，常常用于遥感原始数据的压缩。由于它们多数是有损压缩，一般不用于需要进行分析的栅格数据。在四叉树基础上发展而来的八叉

树目前也是研究热点之一。

- 同所有的数据结构问题一样，压缩编码过程的主要矛盾也是数据量压缩和运算时间之间的矛盾
- 好的压缩编码方法就是要在尽可能减少运算时间的基础上达到最大的数据压缩效率，并且是算法适应性强，易于实现

- 地图代数(map algebra)
  - 对栅格分析所做的大量操作进行组织的一种数学方法
  - 1990年由Tomlin提出
  - 代数包括两个不同元素集合
    - 操作数集合——栅格矩阵
    - 操作集合

- 局部的（local）
- 聚焦的（focal）
- 区域的（zonal）
- 全局的（global）

- 代数需要满足的公理
  - 闭合性，即操作数进行操作的结果必定在操作数集合内。

❖将一个栅格映射到另一个栅格上，新栅格中每个单元格取值依赖于它在原栅格中单元格的值。

❖举例：阈值化

❖新栅格单元格的值依赖于原栅格中相应单元格以及邻近单元格的值。

❖邻域的三种定义

❖新栅格中单元格的值是原栅格中

相应单元格的值以及其他单元格的值的一个函数

- 举例：区域求和

- 新栅格中单元格的值是位置的函数，或者是原栅格或其他栅格上所有单元格的值的函数
- 举例：欧几里得距离

- 专用于图像处理的操作
- 举例：裁剪操作
  - 沿坐标轴提取原栅格的一个子集

- 多边形填充
  - 即在矢量表示的多边形边界内部的所有栅格点上赋以相应的多边形编码
- 主要算法
  - 内部点扩散算法
  - 复数积分算法
  - 射线算法
  - 边界代数算法

- 内部点扩散算法
  - 由多边形一个内部点(种子点)开始，向其8个方向的邻点扩散，判断各个新加入点是否在多边形边界上
    - 如果在边界上，则该新加入点不作为种子点
    - 如果不在边界上，把非边界点的邻点作为新的种子点和原有种子点一起进行新的扩散运算，并标记该种

子点以多边形的编号

- 重复上述步骤直到所有种子点填满该多边形并遇到边界停止为止

❖ 优缺点

- 算法设计复杂
- 受栅格精度限制
  - 复杂图形的同一多边形的两条边界落在同一个或相邻两个栅格内，会造成多边形不连通。

❖ 复数积分算法

- 对全部栅格阵列逐个栅格单元判断该栅格所属的多边形编码
- 判别方法
  - 由待判点对每个多边形的封闭边界计算复数积分
    - 积分值为2∏r，则该点属于此多边形
    - 否则不属于此多边形

❖ 优缺点

- 涉及许多乘除运算，尽管可靠性好，

设计也并不复杂，但运算时间很长，难以在比较低档次的计算机上采用

❖射线算法

- 由待判点向图外某点引射线，判断该射线与某多边形所有边界相交的总次数
  - 偶次：该点在多边形外部
  - 奇次：该点在多边形内部
- 注意特殊情况

❖边界代数算法

- 基于积分思想的转换算法
- 适用于记录拓扑关系的多边形矢量数据转换为栅格结构

❖基本思想

- 提取以相同的编号的栅格集合表示的多边形区域的边界和边界的拓扑关系，并表示由多个小直线段组成的矢量格式边界线的过程。

❖ 步骤

- 多边形边界提取
- 边界线追踪
- 拓扑关系生成
- 去除多余点及曲线圆滑

❖ 多边形栅格转矢量的双边界搜索算法

- 基本思想
  - 通过边界提取，将左右多边形信息保存在边界点上，每条边界弧段由两个并行的边界链组成，分别记录该边界弧段的左右多边形编号
- 步骤
  - 1. 边界点和结点提取

- 2.
- 2.
- 2.
- 2.
- 2.
- 2.边界线搜索与左右多边形信息记录

❖ 网络模型：network model

- 通过目标之间的相关联接
  相互连接多个空间实体

❖ 建模对象

- 交通网、管线网、电力网……

❖ 特性

- 多个要素之间，通过与它们相连接的通道，相互影响和交互
- 可以看成是基于点对象和线对象及其拓扑关系的集合的描述
- 相关现象的精确形状不是非常重要，重要的是具体现象之间的距离或者阻力的度量
- 参与分析运算的对象还包括与网络相关的设施，如商业网点、枢纽、等

- 在GIS中，网络通常用点地物和线地物来建模
  - 点地物
    - 如街道交叉口、电路开关、电闸、水压阀、水流的汇聚点等
  - 线地物
    - 如街道、传送线路、管道、水流分支等
- 模型：有向图结构
  - Digraph = (Vertex, {Relation})
    - Vertex：顶点的集合
    - Relation：两个顶点之间的关系的集合
  - Network = (Node, Arc)
    - Node：结点的集合
    - Arc：弧的集合
  - 结点与弧之间的拓扑关系通过一个连接表来维护

- 矢量数据模型+网络拓扑关系表

- 矢量数据结构：线对象
- 网络拓扑关系表：有向图结构
  - 弧段表AAT

  - 结点表PAT

空间数据模型

主讲：赵 娜

- 1. 空间关系
- 2. 面向对象空间数据模型
- 3. 二维空间对象模型
- 4. 数字表面模型
- 5. 三维空间数据模型
- 6. 网络结构模型
- 7.

- 空间数据模型是地理信息系统的

基础，它不仅决定了系统数据管理的有效性，而且是系统灵活性的关键。

- 空间数据模型是在实体概念的基础上发展起来的，它包含两个基本内容，即实体组和它们之间的相关关系。
  - 实体和相关关系可以通过性质和属性来说明。
  - 空间数据模型可以被定义为一组由相关关系联系在一起的实体集

- 结合空间数据的具体特点进行空间数据模型的设计是地理信息系统的关键。
- 空间数据模型的设计主要是构建一个能够用真实世界的抽象提取来代表该真实世界的模型。

- 由于空间数据模型的设计与计算机硬件、系统软件和工具软件的发展现状密切相关，所以，就目前的发展现状而言，很难用一个统一的数据模型来表达复杂多变的地理空间实体。
  - 例如,某些空间数据模型可能很适合于绘图,但它们对于空间分析来说效率却十分低;有些数据模型有利于空间分析，但对图形的处理则不理想。

- 目前，与GIS设计有关的空间数据模型主要有矢量模型，栅格模型，数字高程模型，面向对象模型，矢量和栅格的混合数据模型等。
  - 前面四种模型属于定向性模型,在模型设计时只包括与应用目标有关的实体及其相互关系

- 混合模型的设计则包括所有能够指出的实体及其相互关系。
- 目前的应用现状而言，矢量模型、栅格模型、数字高程模型相当成熟(目前成熟的商业化GIS主要采用这三类模型)，而其它模型，特别是混合模型则处于大力发展之中。

❖ **空间关系**：描述空间对象之间的空间相互作用关系

❖ **方法**
- 绝对关系：坐标、角度、方位、距离等;
- 相对关系：相邻、包含、关联等
  - 相对关系类型
    - 拓扑空间关系：描述空间对象的相邻、包含等
    - 顺序空间关系：描述空间对象在空间上的排列次序，如前后、左右、东、西、南、北等。
    - 度量空间关系：描述空间对象之间的距离等。

- 地图、遥感影象上的空间关系是通过图形识别的，在GIS中的空间关系则必须显式的进行定义和表达。
- 空间关系的描述多种多样，目前尚未有具体的标准和固定的格式，但基本原理一致。不同的GIS可能采用不同的方法进行描述

- 拓扑关系反映了空间中连续变化的不变性
- 对空间关系的研究主要集中在静态空间二维、三维
  - 三维空间关系极其复杂
- 二维空间拓扑关系
  - 相离、叠加、相接、相等、覆盖、被覆盖、包含，在内部
  - 九交模型

- Egenhofer等人构造了一个由简单几何

实体的边界、内部和外部的点集组成的3×3的矩阵来描述空间拓扑关系，称为9交空间关系模型(9I Model)。该矩阵中的每一元素，都有“空”与“非空”两种取值。

- ❖ 九交矩阵
  - 在这个3*3矩阵中，每个元素根据其交集的空或非空(可以用1和0表示)加以确定
  - 不同的空间关系形成不同矩阵，从而分辨出各种空间关系（理论上讲，可以有2^9=512种关系）

- ❖ 简单面－简单面
- ❖ 简单线—简单线
- ❖ 复杂线—复杂线
- ❖ 线—面

- 优点
  - 简单模型
  - 容易理解，容易实现
- 缺点
  - 不能区分概念上不同的情况

- 采用相离（disjoint）、相等（equal）、相接（touch）、相交（cross）、包含于（in）、包含（contain）、交叠(overlap)、覆盖(cover)、被覆盖(coveredBy)、进入（enter）、穿越（pass）和被穿越（passBy）等共12种基本空间关系表达3D空间中的以下10类有理论价值和实际意义的空间

拓扑关系。

- 描述空间实体的距离或远近等关系。距离是定量描述，而远近则是定性描述。
- 定量描述-距离
  - 欧几里得距离
  - 曼哈顿距离
- 定性描述-远近

- 描述空间实体之间在空间上的排列次序，如实体之间的前后、左右和东南西北等方位关系。
- 在实际应用中，建立和判别三维欧氏空间中的顺序空间关系比二

维欧氏空间中更加具有现实意义。三维欧氏空间中顺序空间关系的建立将为空间实体的三维可视化和虚拟环境的建立奠定必要的技术基础。

- 空间方向关系又称为方位关系、延伸关系，它定义了空间对象之间的方位
- 主要描述为北、东、南、西4个主方向，以及进一步细分东北、东南、西南和西北方向等等。
- 定量表达方式
  - 东北35度
- 定性表达方式
  - 东北、东
- 三种参照方式
  - 内部参照
  - 直接参照
  - 外部参照

- 面向对象（Object Oriented）方法起源于面向对象的编程语言
- 面向对象的方法以对象作为最基本的元素
  - 自然地符合人的认识规律
  - 计算机实现的对象与真实世界具有一对一的对应关系
  - 具有的模块化，信息封装与隐藏、抽象性、多形性等独特之处，为解决大型软件管理，提高软件可靠性、可重用性、可扩充性和可维护性提供了有效的手段和途径

- 对象与封装性
  - 在面向对象的系统中，所有的概念实体都可以模型化为对象。
  - 一个对象是由描述该对象状态的一组数据和表达它的行为的一组操作(方法)组成的。

- 多边形地图上的一个结点或一条弧段是对象，一条河流或一个省也是一个对象。

一个对象object是一个三元组

object = (ID, S, M)

其中，ID为对象标识，M为方法集，S为对象的内部状态，它可以直接是一属性值，也可以是另外一组对象的集合，因而它明显地表现出对象的递归。

❖分类(classification)

- 把一组具有相同结构的实体归纳成类的过程，称为分类
- 这些实体就是属于这个类的实例对象。
- 属于同一类的对象具有相同的属性结构和操作方法。

❖概括(generalization)

- 在定义类型时,将几种类型中某些具

有公共特征的属性和操作抽象出来，形成一种更一般的超类.

- 超类(Superclass)
  - GIS中的地物抽象为点状对象、线状对象、面状对象以及由这三类对象组成的复杂对象。这四种类型就是GIS中各种地物类型的超类
- 子类还可以进一步分类，所以一个类可能是某个或某几个超类的子类，同时又可能是几个子类的超类。
- 建立超类实际上是一种概括，避免了说明和存储上的大量冗余

❖聚集

- 将几个不同特征的对象组合成一个复杂的对象。
  - 每个不同特征的对象是该复合对象的一部分，它们有自己的属性描述数据和操作，这些是不能为复合对

象所公用的

- 但复合对象可以从它们那里派生得到一些信息，它们与复合对象的关系是Parts-of的关系。
  - 例如：房子从某种意义上说是一个复合关系，它是由墙、门、窗和房顶组成的。

- 考虑聚集时，不能强调复合对象的具体细节，每个聚集对象的实例都可以分解成其它成员对象实例，每个成员对象都保持其自有的功能。
- 聚集的操作和其它部分的操作是不兼容的。聚集的每一个操作是由每个部分产生的不同操作所组成的。

❖联合

- 在定义对象时，将同一类对象中的几个具有部分相同属性值的对象组合起来，为了避免重复，设立一个更高水平的对象表示这些相同的属性值。
  - 例如一个县是由若干个乡镇联合而

成。

- 联合常用集合来描述,有联合关系的对象叫成员，所以联合就指的是“成员”关系。
- 在联合中,一个成员对象的具体细节被忽略了，强调的是整个对象的特征。一个集合对象的实例可以分解成一系列其成员对象的实例。联合通过其成员产生集合数据结构,一个集合的操作是由该集合每个成员的操作组成的。

❖ 继承

- 服务于概括
- 减少代码冗余,减少相互间的接口和界面

❖ 传播

- 作用于聚集和联合的工具,用于描述

复合对象对成员对象的依赖性并获得成员对象的过程。

- ❖ 二维基本空间对象模型
  - 空间地物几何数据模型
- ❖ 拓扑关系与面向对象模型
- ❖ 面向对象的属性数据模型
- ❖ OGC
  - http://www.opengeospatial.org/
  - OGC全称Open Geospatial Consortium，非盈利的、国际化的、自愿协商的标准化组织
  - 主要目的就是制定与空间信息、基于位置服务相关的标准。这些标准就是OGC的“产品”，而这些标准的用处就在于使不同厂商、不同产品之间可以通过统一的接口进行互操作。
  - 在GIS领域，OGC已经是一个比较“官方”的标准化机构了，它不但包括了ESRI、Google、Oracle等业界强势企业作为其成员，同时还

和W3C、ISO、IEEE等协会或组织结成合作伙伴关系。因此，OGC的标准虽然并不带有强制性，但是因为其背景和历史的原因，它所制定的标准天然地具有一定的权威性。

- 使用关系-对象数据库技术
- 空间数据对象及其相互间的关系、 使用和连接规则等均可以方便地表示、存储、管理和扩展 。
- 引入这种新的数据模型的目的在于让用户可以通过在他的数据中加入其应用领域的方法或行为以及其他任意的关系和规则，使数据更具智能和面向应用领域。

- Geodatabase结构

- 要素类（Feature class）
  - 同类空间要素的集合即为要素类。如:河流、道路、电缆等。

- 要素数据集（Feature dataset）由一组具有相同空间参考（Spatial Reference）的要素类组成。
  - 专题归类表示
    - 当不同的要素类属于同一范畴（如水系的点线面要素）
  - 创建几何网络
    - 在同一几何网络中充当连接点和边的各种要素类（如配电网络中，有各种开关、变压器、电缆等）
  - 考虑平面拓扑
    - 共享公共几何特征的要素类(如:水系、行政区界等）

- 关系类（Relationship class）
  - 定义两个不同的要素类或对象类之

间的关联关系　如:我们可以定义房主和房子之间的关系

- ❖几何网络（Geometric network）
  - 几何网络是在若干要素类的基础上建立的一种新的类。定义几何网络时，指定哪些要素类加入其中，同时指定其在几何网络中扮演什么角色
  - 如: 定义一个供水网络，指定同属一个要素数据集的“阀门”、“泵站”、“接头”对应的要素类加入其中，并扮演“连接（junction）”的角色;同时，指定同属一个要素数据集的“供水干管”、“供水支管”和“入户管”等对应的要素类加入供水网络，由其扮演“边（edge）”的角色。
- ❖域（Domains ）
  - 定义属性的有效取值范围。可以是连续的变化区间，也可以是离散的取值集合。

- 有效规则（Validation rules）
  - 对要素类的行为和取值加以约束的规则。如：规定不同管径的水管要连接，必须通过一个合适的转接头。规定一块地可以有一到三个主人。
- 栅格数据集（Raster Datasets）
  - 用于存放栅格数据。可以支持海量栅格数据，支持影像镶嵌，可通过建立“金字塔”索引，并在使用时指定可视范围提高检索和显示效率。
- TIN Datasets
- Locators

- Geodatabase的拓扑关系规则
  - 拓扑关系规则可作用于同一要素数据集中的不同要素类或者同一要素类中的不同要素。用户可以指定空间数据必须满足的拓扑关系约束，譬如：要素之间的相邻关系、连接关系、

覆盖关系、相交关系、重叠关系等。所有这些关系都对应相应的规则。

- 在城市规划应用中，两个相邻的地块之间不能有“飞地”，可以有一条对应的规则：“相邻多边形间不能存在间隙”。再如，当以河流作为国界时，河流（线状）与国界线必须一致，可用规则：“线必须被多边形边线覆盖”。

❖Geodatabase的优势

- 在同一数据库中统一管理各种类型的空间数据
- 空间数据的录入和编辑更加准确
- 空间数据更加面向实际的应用领域
- 可以表达空间数据的相互关系
- 可以更好的进行制图
  - 对不同的空间要素，可定义不同的

“绘制”方法，而不受限于ArcInfo等客户端应用已经给出的工具

- 空间数据的表示更加准确
  - 除了可用折线方式以外，还可用圆弧、椭圆弧和Bezier曲线描述空间数据的空间几何特征。
- 可管理连续的空间数据
- 支持空间数据的版本管理和多用户并发操作

❖数字高程模型(DEM, Digital Elevation Model)

- 采用规则或不规则多边形拟合面状空间对象的表面,主要是对数字高程表面的描述。
- 根据多边形的形状,把数字高程模型分为两种,即不规则三角网模型和网格模型。

❖数字高程模型的主要优点是能够方便地进行空间分析和计算，如

对地表坡度、坡向的计算等。

- ❖利用不规则三角形来描述数字高程表面。
- ❖在TIN模型中，可以建立三角形顶点（数据点）、三角形边、三角形个体间的拓扑关系。
- ❖如果建立了TIN模型图形实体（三角形顶点、三角形边、三角形）的拓扑关系，将大大加快处理三角形的速度。

- ◆TIN是使用彼此相邻而不重叠的三角形组成的三角网，每个三角形顶点的xyz坐标已知，所以通过在一个三角形表面使用简单的线性插值和多项式插值，可以估计任何位置的表面值；
- ◆TIN的基本组成是三角形（Triangles），而三角形由

节点（Nodes）和边(Edge)。Nodes 是由x,y,z定义的坐标和变量值组成，边Edges 即指三角形的边。三角形Triangles由节点按一定规则相连形成的；

◆TIN不但由连续点组成，也可包含突变或断线（表示为三角形的边）。

- 1.
- 点（Mass Point）：最终形成三角形的节点。
- 2.
- 线(Breakline)：最终形成三角形的边，通常表示地理现象大的转折点
- 3.
- 多边形(Clip Edge)：对多边形以外的区域不予考虑。

- 1.侧重点：TIN 模型侧重于面的三角部分，回答某个三角区域的表面像什么；而栅格模型侧重于单个单元格位置，回答每个单元格代表的区域是什么；
- 2.形状：TIN 能精确地表示曲面类型地理现象的形状；而栅格模型不能精确表示；
- 3.空间对象的表示：栅格数据模型可以表示离散的地理现象，而TIN模型只能表示联续变化的地理现象；两种模型都可表示曲面上的地理现象的渐变；
- 4.表面模型的精度，栅格使用统一的CELL大小来表示，CELL越小，精度越高，而TIN具有随坡度变化而不同的点密度，在坡度变化大的地区点密度较高；栅格模型不能精确定位如山脊、山峰等，而TIN模型中这些典型地形特征被特别存储，位置和表面值都很精确；
- 5.栅格模型适合进行空间一致性分析、近邻分析、离散度分析及表面最低成本分析，TIN模型适合进行坡度、坡向、体积计算和视线分析等

- 与栅格模型相似，同样是直接采用面域或空域枚举来描述空间目标对象。
  - 栅格模型的每一像元或像元的中心点代表一定面积范围内空间对象或实体的各种空间几何特征和属性几何特征，而网格模型通常以行列的交点特征值代表交点附近空间对象或实体的各种空间几何特征和属性几何特征。
  - 栅格模型主要用于图象分析和处理，而网格模型主要进行等值线的自动生成，坡度、坡向的分析等。
  - 栅格模型处理的数据主要来源于航空、航天摄影以及视频图象等，而网格模型则主要来源于原始空间数据的插值。

❖空间数据的三维特征

- 几何坐标增加了第三维信息
- 空间拓扑关系复杂化
- 三维地理空间中的三维对象具有丰富的内部信息

❖三维数据结构

- 最简单的方法：采用三维行程编码
- 复杂方法：八叉树三维数据结构
  - 用八叉树来表示三维形体。
  - 节点类型
    - 灰节点：对应立方体部分地被占据
    - 白节点：对应立方体没有被任何物体占据
    - 黑节点：对应立方体全部被物体占据

❖网络模型

- 如果取消层次模型中的二个限制，即每一个结点可以有多个父结点，便形成了网络，又称为丛。我们把用丛结构来表示实体之间联系的模型叫网络模型。

空间数据组织与管理

主讲：赵 娜

❖GIS定义？

- 地理信息系统（Geographical Information System, GIS）是一种特定的十分重要的空间信息系统，它是在计算机硬件、软件系统支持下，对整个或部分地球表层（包括大气层）空间中的地理分布数据进行采集、储存、管理、运算、模拟、分析、显示和描述的技

术系统。

- ❖ 矢量数据模型与栅格数据模型
  - 定义、特点、结构与编码方法

- ❖ 存储到哪里？
- ❖ 以何种方式来存储？？？
- ❖ 如何来管理？？？

- ❖ 1. 文件组织与数据库
- ❖ 2. 空间数据管理方式
- ❖ 3. 空间数据引擎
- ❖ 4. 空间数据与属性数据的连接
- ❖ 5. 空间数据组织
- ❖ 6. 栅格数据存储与管理

- ❖ 数据库中数据组织层次
  1. 数据项(元素/基本项/字段)：定义数据的最小单位

2. 记录：由若干相关联的数据项组成。
3. 文件：一给定类型的记录的全部具体值的集合。
4. 数据库：若干文件的集合。

数据库是具有特定联系的数据的集合，也可看成是具有特定联系的多种类型的记录的集合。

❖数据库管理系统（DBMS）

- 位于用户和操作系统之间进行数据库存取和各种管理控制的软件,使数据库系统的中心枢纽,在用户应用程序和数据文件之间起到桥梁的作用
- DBMS软件
  - Oracle
  - SQL Server
  - Access

❖空间数据库系统

- 定义
  - 带有数据库的计算机系统，采用现代数据库技术来管理空间数据。
- 广义地理解为
  - 包括空间数据库本身，计算机软硬件，空间数据管理系统，地理空间数据库和数据库管理人员一整套运行系统
- 目的
  - 有效组织空间数据
  - 建立空间索引，快速调度任意范围的数据

❖ 矢量数据管理

- 1. 文件管理
- 2. 文件与关系数据库混合管理
- 3. 全关系型空间数据库管理
- 4. 面向对象空间数据库管理
- 5. 对象-关系数据库管理

❖ 栅格数据管理

- 对于矢量数据，其位置数据和属性数据通常是分开组织的。
  - 1. 文件管理
  - 2. 文件与关系数据库混合管理
  - 3. 全关系型空间数据库管理
  - 4. 面向对象空间数据库管理
  - 5. 对象-关系数据库管理
  - 6.

- 各个地理信息系统应用程序对应各自的空间和属性数据文件，当两个GIS应用程序需要的数据有相同部分时，可以提出来作为公共数据文件。
- GIS软件：MapInfo
- 缺点：

  1）程序依赖于数据文件的存储结构，

数据文件修改时，应用程序也随之需要改变。

2）以文件形式共享，当多个程序共享一数据文件时，文件的修改，需得到所有应用的许可。不能达到真正的共享，即数据项、记录项的共享。

- 两个子系统分别存储和检索空间数据和属性数据，使用一种标识符将两者联系起来
- 属性数据建立在RDBMS上，数据存储和检索比较可靠、有效；
- GIS软件：Arc/Info，MGE，SICARD、GENEMAP等。

- 缺点：
  - ①属性数据和图形数据通过ID联系起来，使查询运算，模型操作运算速度慢；
  - ② 数据分布和共享困难；
  - ③属性数据和图形数据分开存储，数据的安全

性、一致性、完整性、并发控制以及数据损坏后的恢复方面缺少基本的功能;

④缺乏表示空间对象及其关系的能力。

- 图形数据与属性数据都采用关系型数据库存储。
- 本质:
  - GIS软件商在标准DBMS顶层开发一个能容纳、管理空间数据的系统功能。
- 特点:
  - 空间数据和属性数据不必进行烦琐的连接,数据存取较快
  - 属性间接存取效率比DBMS的直接存取慢,特别是涉及空间查询、对象嵌套等复杂的空间操作

- GIS软件: System9, Small World, Geovision等

- 面向对象的空间数据库管理系统允许

用户定义对象和对象的数据结构及操作。这样，我们可以将空间对象根据GIS的需要，定义出合适的数据结构和一组操作。

- 面向对象模型最适应于空间数据的表达和管理，它不仅支持变长记录，而且支持对象的嵌套、信息的继承与聚集。
- 当前已经推出了若干个面向对象数据库管理系统，也出现一些基于面向对象的数据库管理系统的地理信息系统，但由于面向对象数据库管理系统还不够成熟，价格又昂贵，目前在GIS领域还不太通用。
- 相反基于对象--关系的空间数据库管理系统是目前GIS空间数据管理的主流。

- 扩展的空间对象管理模块主要解决了空间数据的变长记录的管理，由于由数据库软件商进行扩展，效率要比前面所述的二进制块的管理高得多。

- 许多数据库管理系统的软件商纷纷在关系数据库管理系统中进行扩展，使之能直接存储和管理非结构化的空间数据，如Oracle 和Informix等都推出了空间数据管理的专用模块，定义了操纵点、线、面、圆、长方形等空间对象的API函数。用户不能根据GIS要求对其进行再定义，一般不带拓扑关系。
- 但是它仍然没有解决对象的嵌套问题，空间数据结构也不能由用户任意定义，使用上仍然受到一定限制。
- GIS软件：TIGER，Geo++、Geo Tropics等

Oracle Spatial是什么？

- 一系列函数和过程的集合，在Oracle9i数据库中实现了对空间信息的存储、访问和分析
- 提供了SQL模式和函数来实现Feature Collection的存储、检索、更新和查询

Oracle Spatial的组成

- 实现模式（MDSYS）
  - 规定了支持的几何数据类型的存

储、语法和语义

- 空间索引机制

❖Oracle Spatial的组成

- 一套运算符和函数
  - 进行感兴趣区域查询，空间连接查询和其它空间分析操作
- 管理工具

❖对象关系模型Object Relational Model

- Spatial采用对象关系模型表示几何对象
  - 定义类型为MDSYS.SDO_GEOMETRY的字段
  - 每个几何对象无需占用多行存储
  - 对应OpenGIS Feature实现规范中的“SQL92+Geometry” Feature实现方案

❖对象关系模型的优势

- 支持丰富的几何对象类型
  - 包括圆弧arc ，圆circle, 混合多边形compound polygon, 混合折线段compound line string, 以及优化的矩形
- 易于创建和维护空间索引以及构造空间查询
- 空间索引由Oracle9i数据库服务器自动维护
- 几何对象可以保存在单条记录的单个字段
- 优化的性能

❖ Spatial支持的几何对象类型
- Point, point cluster
- Line string
- Polygon
- Arc line string
- Arc polygon
- Compound polygon
- Compound line string
- Circle

- Rectangle

❖Spatial支持3维和4维几何对象类型
- 仅实现存储和索引
- 空间函数仅对前两维坐标操作
- 空间运算符对多于两维的对象无效

❖Spatial的层次结构数据模型
- 高层次的对象由低一层次的对象构成
- 包括元素element，几何形geometry和图层layer三个层次

❖定义

❖SDO_GTYPE
- 指明了geometry的类型，对应OGIS几何对象模型中的类型
- 是一个4位数字，格式为dltt，其中：

- d代表geometry的维数:2,3,4
- l代表线性参照系LRS测量维度，非LRS为0
- tt指明了geometry的类型，00-07，08-99保留

- geometry的几个方法可以返回类型值
  - GET_DIMS, GET_LRS_DIM, GET_GTYPE

❖SDO_GTYPE

❖SDO_SRID

- SDO_SRID用来确定geometry所对应的空间参照系
  - null代表不对应任何参照系
  - 非空值必需参照MDSYS.CS_SRS表的SRID字段，同时需要存入USER_SDO_GEOM_METADATA表的SRID字段

- 在同一个字段里的所有geometry对象都必须具有相同的SRID

❖ SDO_POINT
- 为了性能优化而设置的
- 如果SDO_ELEM_INFO和SDO_ORDINATES成员都为空的话，SDO_POINT就是一个POINT对象的坐标值，否则SDO_POINT被忽略
- 如果图层中仅有POINT对象的话，强烈建议将坐标值存放在SDO_POINT中

❖ SDO_ELEM_INFO
- 是一个变长的数值数组，说明SDO_ORDINATES成员中数值的意义
- 每三个数值为一个单元，分别是:
  - SDO_STARTING_OFFSET: 元素坐标值从SDO_ORDINATES数组第几个数开始，>=1
  - SDO_ETYPE: 说明元素的类型，1/21003/2003表明是一个简单元素;

4/1005/2005表明是一个复合元素

- SDO_INTERPRETATION: 当ETYPE=1003/2003，指明解释SDO_ORDINATES的方式；当ETYPE=1005/2005,指明后面的几个单元构成复合元素

❖SDO_ORDINATES

- 存放坐标值，如何解释由SDO_ELEM_INFO来指明

❖有效性约束

- GTYPE=d001/d005(点)，则ETYPE=1
- GTYPE=d002/d006(线)，则ETYPE=2/4
- GTYPE=d003/d007(面)，则ETYPE=3/5/1003/2003/1005/2005

❖Oracle Spatial空间索引技术

❖Oracle Spatial提供了R树索引和

四叉树索引两种索引机制来提高空间查询和空间分析的速度。用户需要根据空间数据的不同类型创建不同的索引，当空间数据类型比较复杂时，如果选择索引类型不当，将使Oracle Spatial创建索引的过程变得非常慢。

- 影像数据和数字高程模型（Digital Elevation Model，DEM）
- 金字塔结构存放多种空间分辨率的栅格数据
  - 越靠近顶层，数据的分辨率越小，数据量也越小，只能反映原始数据的概貌
  - 越靠近底层，数据的分辨率越大，数

据量也越大，更能反映原始详情

❖组织形式

- 栅格目录
  - 管理具有相同空间参考系的多幅栅格数据，各栅格数据在物理上独立存储，易于更新
  - 用于管理更新周期快，数据量大的影像数据
- 栅格数据集
  - 管理具有相同空间参考系的一幅或多幅镶嵌而成的栅格影像数据，物理上实现数据的无缝存储
  - 适合管理DEM等空间连续分布、频繁用于分析的栅格数据类型

❖存储结构

- 金字塔层
- 波段
- 数据分块

- 三种方式
  - 基于文件的影像数据库管理
  - 文件结合数据库影像管理
  - 基于关系数据库管理

- 目前大部分GIS软件和遥感图像处理软件都是采用文件方式来管理遥感影像数据。
- 缺点：
  - 大量的图像元数据信息（如图像类型、摄影日期、摄影比例尺等）需要单独建立文件；
  - 多数据源、多时相的遥感图像数据间的关系无法反映；
  - 数据的安全性、并发控制和数据共享问题。

- 影像数据仍按照文件方式组织管理；在

关系数据库中，每个文件都有唯一的标识号（ID）对应影像信息，如文件名称、存储路径等

表1 影像信息数据库表

影像名称	块号	…
Image 001	011001	…
Image 002	011002	…
Image 003	011003	…
Image 004	021001	…
Image 005	021002	…
… …	… …	…

- 基于扩展关系数据库的影像数据库管理是将影像数据存储在二进制变长字段中，然后应用程序通过数据访问接口来访问数据库中的影像数据。同时影像数据的元数据信息也存放在关系数据库的表中，二者可以进行无缝管理。

- 数据库方式管理影像数据具有以下特点：
  ① 所有数据集中存储，数据安全，易于共享。
  ② 较方便管理多数据源和多时态的数据。
  ③ 支持事务处理和并发控制，有利于多用户的访问与共享。
  ④ 影像数据和元数据集成到一起，能方便的进行交互式查询。
  ⑤ 对Client/Server的分布式应用支持较好，网络性能和数据传输速度都有很大提高。
  ⑥ 影像数据访问只能通过数据库驱动接口访问，有利于数据的一致性和完整性控制，数据不会被随意移动、修改和删除。
  ⑦ 支持异构的网络模式，即应用程序和后台数据库服务器可以在不同操作系统平台下运行。

- 地理空间数据库引擎技术的目的
  - 需求：
    - 空间-属性数据一体化
    - 矢量-栅格数据一体化
    - 空间信息-业务信息一体化

- 目的:
  - 通过空间数据库引擎，可以用传统的关系数据库对空间地理数据加以管理和处理，提供必要的空间关系运算和空间分析功能。
  - 实现客户/服务器的分布计算模式，实现地理空间数据的透明访问、共享和互操作，从而建立真正意义上的分布式空间地理数据库。

❖1996年，ESRI公司与Oracle等数据库开发商合作，开发出一种能将空间图形数据也存放到大型数据库中管理的产品，将其定名为“spatial database engine”，简称SDE，即为空间数据库引擎

❖扩展关系数据库管理系统管理地

理空间数据的两种途径

- 寄生在关系数据库管理系统之上的空间数据引擎
  - ESRI公司的ArcSDE，MapInfo公司的Spatial Ware
  - ✓支持通用的关系数据库管理系统，空间数据按BLOG存，可以跨数据库平台，与特定GIS平台结合紧密
  - ×空间操作与处理无法在数据库内核中实现，数据模型复杂，扩展SQL困难直接扩展通用数据库的空间数据库系统
  - Oracle Spatial
  - ✓空间数据的管理与通用数据库系统融为一体，空间数据按对象存取，可在数据库内核实现空间操作和处理，扩展SQL比较方便，易实现数据共享与互操作
  - ×实现难度大，压缩数据比较困难

- ESRI对SDE定义：从空间数据管理的角度来看，SDE可看成是一个连续的空间数据模型，借助这一模型，可将空间数据加入到关系数据库管理系统 (RDHMS)中去。它允许向关系数据库中加人空间数据、提供地理要素的空间位置及形状等信息。

- 其它定义1:
- 空间数据引擎，简称SDE，是一种空间数据库管理系统的实现方法，即在常规数据库管理系统之上添加一层空间数据库引 擎，以获得常规数据库管理系统功能之外的空间数据存储和管理 的能力。

- 其它定义2:
  - 空间数据引擎（Spatial Data Engine）是运用统一的数据接口来管理不同空间数据源中的空间数据，解决不同格式的空间数据与应用程序之间的数据接口问题。即空间数据引擎是解决空间数据对象中几何和属性信息在不同数据源中的存取问题，其主要任务是:
    - 用多种数据源存储管理空间数据，包括File、DBMS和XML
    - 从数据源中读取空间数据，并转换为GIS 应用程序能够接收和使用的格式
    - 将GIS 应用程序中的空间数据导入各种数据源，并进行管理
- 因此，空间数据引擎直接为程序提供了空间数据服务，是空间数据进出各数据源的通道。

- 综合定义:
- SDE可以理解为基于特定的空间数据模型，在特定的数据存储、管理系统的基础上（典型的是数

据库管理系统)，提供对空间数据的存储、检索等操作，以提供在此基础上二次 开发的程序功能集合。

- 将空间数据类型加到关系数据库中，不改变和影响现有的数据库和应用，只是在现有的数据表中加入图形数据项，供软件管理和访问与其相关联的空间数据，SDE通过将信息存入层表来管理空间可用表。

（1）与空间数据库联合，为任何支持的用户提供空间数据服务。
（2）提供开放的数据访问，通过TCP/IP横跨任何同构或异构网

络， 支持分布式的GIS系统。

（3）SDE对外提供了空间几何对象模型，和户可以在此模型基础之上建立空间几何对象，并对这些几何对象进行操作。

（4）快速的数据提取和分析。SDE提供快速的空间数据提取和分析功能，可进行基本拓扑的查询、缓冲区分析、叠加分析、合并和切分等。

（5）SDE提供了连接DBMS数据库的接口，其他的一切涉及与DBMS 数据库进行交互的操作都是在此基础之上完成。

（6）与空间数据库联合可以管理海

量空间信息，SDE在用户与物理数据的远程存储之间构建了一个抽象层，允许用户作逻辑层面上与数据库 交互，而实际的物理存储则交由数据库来管理。海量的数据是由空间数 据库管理系统来保障的。

（7）无缝的数据管理，实现空间数据与属件数据统一存储。传统的地理信息存储方式是将空间数据与属性数据分别存储，空间数据因其复杂 的数据结构，多以文件的形式保存，而属性数据多利用关系数据库存 储。而SDE涉及空间属性数据在DBMS中如何存储及管理，通过SDE则可以把这两种数

据同时存储到数据库中，实现空间属性数据一体化管 理，保证了更高的存储效率和数据完整性。

（8）并发访问。SDE与空间数据库相结合，提供空间数据的并发响应机制。用户对数据的访问是动态的、透明的。

❖SDE，一方面可以实现海量数据的多用户管理、数据的高速提取和空间分析，以及同开发环境良好的集成和兼容，同应用系统无缝嵌 入；另一方面屏蔽掉了不同数据库和不同GIS文件格式之间的壁垒，实现了多源数据的无缝集成，从而为最终实现GIS的互操作

提供一种有效途径。

- MapInfo公司的Spatial Ware
- ArcGIS空间数据库引擎
- 国内SuperMap公司的XSDE

MapInfo SpatialWare

- MapInfo Spatialware是MapInfo公司推出的空间数据库服务器，目前已发布了基于DB2、MS SQL Server、Informix数据库的各种版本。

❖第一个在对象-关系型数据库环境中基于SQL进行空间查询和分析的空间信息管理系统。

MapInfo SpatialWare

--功能特点

❖支持广泛的、多用户的、分布式的地图应用程序。

★ 支持海量数据，将空间数据与商业数据无缝的结合在一起。

★ 支持Server端的数据处理。

★ 空间索引机制，加快数据检索。

★ 扩展的SQL/Spatial语言，加快数据库端的查询和分析。

- ArcSDE采用C/S结构
- ArcSDE是中间件
  - 在关系数据库管理系统（RDBMS）中存储和管理多用户空间数据库的通路
  - ArcSDE提供对空间和非空间数据进行高效率操作的数据库服务
  - ArcSDE还提供了应用程序接口，软件开发人员可以把空间数据检索和分析功能集成到自己的应用程序中

- 主要功能与特点
  - 1.高性能的DBMS通道
  - 2.开放的DBMS支持
  - 3.多用户
  - 4.连续、可伸缩的数据库
  - 5.GIS工作流和长事务处理
  - 6.丰富的地理信息数据模型
  - 7.灵活的配置

- SuperMap SDX+的特点
  - 基于大型商用数据库
  - 管理空间数据及属性数据
- SDX+的数据组织
  - 空间与属性一体化存储
  - 数据组织形式
  - 系统表
  - 使用RDBMS管理数据
- SDX+的索引技术

- 数据源 - DataSource
- 数据集 - Dataset
  - 矢量数据集 - DatasetVector

- 栅格数据集 - DatasetRaster

❖记录集 - Recordset

❖记录 - Record

❖技术特点

- 支持多种数据库
- 支持矢量和栅格数据
- 支持存储拓扑关系
- 提供长事务处理能力
- 采用三级索引技术
- 可选的文件缓存
- 支持矢量数据有损/无损压缩
- 支持影像压缩

- 时序数据支持
- 编辑操作性能提升

❖ 图形数据与专题属性数据分别管理
  - 属性数据作为图形数据记录的一部分进行存储
    - 当属性数据量不大时使用
  - 用单向指针指向属性数据
    - 缺点是仅有从图形到属性的单向指针

❖ 对通用DBMS扩展以增加空间数据的管理能力

❖ 属性数据与图形数据具有统一的结构

❖ 图形数据与属性数据自成体系

- 纵向分层组织
- 横向分块组织
- 分层分块索引
- 三维空间数据组织

空间数据索引技术

- 索引问题的提出
  - 假想在一个没有进行任何管理的图书馆中索取一份自己想要的资料,在一个没有字母索引的字典里查找生字
  - 为了避免这种毫无方向漫无边际的检索必须提出一种能加快定位速度的有效方法，于是索引技术应运而生。
  - 给一个庞大的数据集找到一个有效的索引体系是十分重要的,特别对于空间数据这种海量数据而言更是如此。所以有这样的说法“海量数据如

无索引管理将寸步难行，必将成为‘数据坟墓’,丢不敢丢,用不能用”。

- 一个信息系统不论是一般的关系型数据库还是空间数据库,其一项根本的任务就是信息的检索查询。能否快速的检索信息是数据库性能高低的一个主要的标志。

❖数据索引问题的提出

- 由于计算机的体系结构将存贮器分为内存、外存两种，访问这两种存储器一次所花费的时间一般为30¯40ns，8¯10ms，可以看出两者相差十万倍以上
- 绝大多数数据是存储在外存磁盘上的,如果对磁盘上数据的位置不加以记录和组织,每查询一个数据项就要扫描整个数据文件,这种访问磁盘的代价就会严重影响系统的效率

- 因此系统的设计者必须将数据在磁盘上的位置加以记录和组织，通过在内存中的一些计算来取代对磁盘漫无目的的访问，才能提高系统的效率

❖ 传统的索引方法只适用于一维序集

❖ 研究空间数据索引的意义

- 当工作区数据量较大，特别是用无缝的空间数据库管理整个工程的空间数据时，要查询鼠标到底圈选了哪些地物？

❖ 空间索引的定义

- 在存储空间数据时依据空间对象的位置和形状或空间对象之间的某种空间关系，按一定的顺序排列的一种数据结构，其中包含空间对象的概要信息，如对象的标识、外接矩形及指向空间对象实体的指针。

❖ 空间数据库系统的响应时间主要

由数据的定位时间和数据的提取时间来决定。

- 查询时间主要消耗在数据定位上,即空间索引的时间
  - 空间索引性能的优劣直接影响空间数据库和地理信息系统的整体性能。
- 数据提取时间与代提取数据的规模成正比

❖传统的数据库索引技术

- B树、B+树、二叉树、ISAM索引、哈希索引
- 针对一维属性数据的主关键字索引而设计
- 不能对空间数据进行有效的索引

❖空间索引技术

- 利用某种空间联系组织数据项,数据项的码值可以看成是k维空间中的一个点

- 传统索引（以B+树为例）
  - 存储方式：基于值的二维空间线性化存储
  - 适用查询:只能适用于针对某一字段的查询
- 空间索引
  - 存储方式：基于邻近度存储数据项
  - 适用查询：“找出一个给定点的10个最邻近的邻居”,“找出与一个给定点相距特定距离的所有点”
  - 缺点:如果所有的数据项都按照某一字段顺序排序,则空间索引比以这个字段为搜索码的第一个字段的B+树索引慢

- 空间索引的研究始于20世纪70年代中期
  - 早于空间数据库研究
  - 初始目的：提高多属性查询效率
  - 主要研究检索多维空间点的索引,逐渐扩展到其他空间对象的检索

- 空间索引发展历程

- 空间索引技术
  - 树结构
  - 线性映射
  - 多维空间区域变换
- 从应用范围上分为静态索引和动态索引

- 5.2.1网格索引原理
- 5.2.2网格索引算法
- 5.2.3简单网格索引编码

- 思路：

  将研究区域用横竖线条划分大小相等或不等的格网，记录每一个格网所包含的空间实体。当

用户进行空间查询时，首先计算出用户查询对象所在格网，然后再在该网格中快速查询所选空间实体，这样一来就大大地加速了空间索引的查询速度。

- 网格化可以通过使网格编号向正负方向上不断延展以反映整个二维空间的情况
- 网格索引在追加新的实体记录时不论在扩展网格范围还是增加网格记录上都具有很高的扩展性

- 评价网格索引性能的指标
  - 网格大小

- 网格索引表记录数
- 网格索引表记录数与实体记录数的比率
- 平均每格的实体数
- 最大每格的实体数
- 完全分布在一个网格中的实体百分比

❖ 网格大小，制约和影响其它指标

- 网格越大，网格索引表记录数越小，越与实体记录数相接近,进而影响网格索引表记录数与实体记录数的比率,但平均及最大每格实体数也越多
- 网格越小，网格索引表记录数越多，但平均及最大每格实体数相对变少，完全分布在一个网格中的实体百分比也会降低
- 理想的网格大小是使网格索引表记录不至于过多,同时使平均及最大每格的实体数处于较低水平

- 关键操作
  - 创建
  - 重建索引
  - 查询
  - 插入
  - 删除和更新

- （1）传统单元网格索引编码
  - 在建立地图数据库时需要用一个平行于坐标轴的正方形数学网格覆盖在整个数据库数值空间上，将后者离散化为密集栅格的集合，以建立制图物体之间的空间位置关系。
  - 通常是把整个数据库数值空间划分成32×32（或64×64）的正方形网格，建立另一个倒排文件——栅格索引。

- 每一个网格在栅格索引中有一个索引条目(记录)，在这个记录中登记所有位于或穿过该网格的物体的关键字。
  - 变长指针法
  - 位图法

❖利用传统单元网格索引查询的SQL模型如下:　(GCODE为网格编码列名称)

```
SELECT id0 from owner.GeoObjTb1 WHERE
  GCODE IN (...)
```

❖传统简单网格索引特点

- 编码过于简单,使得编码值在空间上不能保持连续性

❖ (2) 改进型单元网格索引编码

- 改进型单元网格索引将传统型编码

由1维升至2维，变成X和Y方向上的编码;

- 将空间要素的标识、空间要素所在的网格的X和Y方向上的编码、以及空间要素的外包络矩形作为一条数据库记录存储。
- 如果一个空间要素跨越多个网格,则同样存储多条记录。

❖ 特点

- 保持了网格在空间上相邻则编码值也相邻的特性
- 构造查询的SQL模型时，可以使用连续表示方式
- 该索引存储了空间要素的外包络矩形,可以为查询过滤非查询范围的要素提供进一步依据

❖ 假设给定查询范围为矩形，坐标为

gxmin, gymin, gxmax, gymax，其网格编码范围为：X方向(gcodexmin—gcodexmax),Y方向（gcodeymin－gcodeymax），gcodex和gcodey分别是网格编码列名称。查询矩形范围相交的空间要素（包括矩形范围内的空间要素和交到了矩形范围边界的空间要素）

- 利用单元网格索引过滤的SQL模型如下：

```
SELECT id0 from owner.GeoObjTbl
WHERE （gcodex>= gcodexmin AND
gcodex<=gcodexmax AND gcodey>=
gcodeymin AND gcodey<=gcodeymax）
AND (((gxmin<=XMAX AND gxmin >=XMIN)
OR (gxmax<=XMAX AND gxmax>= XMIN))
AND ((gymin<=YMAX AND gymin> =YMIN)
OR (gymax<= YMAX AND gymax>=YMIN)))
```

- 索引工作机制如下：
- SDE在执行该空间查询时，从客户

端接收查询多边形的外包络矩形以及查询多边形外包络矩形所跨的网格单元，这些信息通过SQL模型的WHERE子句传递给DBMS。

- 第一步，DBMS从SDE接收SQL语句（该语句包括网格单元和外包络矩形的坐标）。WHERE子句定义了在空间索引中需要选择的网格单元。一旦在空间索引表中确定了网格单元，外包络矩形的搜索就从所选择的网格边界开始;
- 第二步，利用查询多边形的外包络矩形和空间索引表中的空间要素的外包络矩形，DBMS可减少最初的选择集。DBMS比较查询多边

形的外包络矩形和空间要素的外包络矩形是否有重叠，如果有，则该空间要素被选择来做下一步的空间要素边界比较，形成一个最初选择集;

- 第三步，在SDE中，用查询多边形的外包络矩形与最初选择集中的空间要素的边界坐标进行比较，如果查询多边形的外包络矩形与第二步选择集中的空间要素边界不重叠，该空间要素就从最初选择集中过滤掉，结果形成中间选择集;
- 第四步，将查询多边形的边界坐标和中间选择集的空间实体边界坐标进行比较，一旦有重叠发生，比较的结果记录就形成最终结果集。该步比较过程是一个二进制比较过程，将花费较多的时间和空间。(如果查询多边形为矩形，则

该步操作可省略，中间选择集直接升级为最终结果集)。

- 第一步和第二步是用来减少由SDE执行的空间要素边界比较的次数，减少返回的数据记录有助于减少空间要素比较的数量，因而可以缩短空间查询的时间。

- 网格单元大小因素
- 单元网格索引是一种多对多的关系，即一个网格单元可以包含多个空间要素，且一个空间要素可以跨越多个网格单元。在这种多对多的关系下，网格的大小是影响索引效率的最主要因素。

- 与空间要素的外包络矩形大小相比，网格单元很大时，将导致每

个网格单元内包含有很多空间要素。第一阶段选择的网格虽少，但导致第二阶段将不得不处理大量网格内的空间要素的边界比较，潜在地增加了查询的时间。

- 如果网格单元太小，小于空间要素外包络矩形的平均大小，将会导致空间索引表产生大量的网格单元，并且很多网格单元都索引出相同的空间要素。当大量的空间要素外包络矩形被网格单元切割时，空间索引表变大，因而查询网格单元时间增长。

- 网格单元的大小不是一个确定性的问题，需要多次尝试和努力才会得到好的结果。有一些确定网格初始值的原则，

用它们可以进一步确定最佳网格大小，可在任何时候重新计算网格的大小，使DBMS重建空间索引表。如果空间要素外包络矩形的大小变化比较大，可以选择多种网格大小，但在空间索引搜索的过程中DBMS必须搜索所有网格单元级，这将消耗大量时间。

- 最佳网格的大小可能受图层平均查询的影响，如果用户经常对图层执行相同的查询，经验数据表明，网格的大小为查询空间范围的1.5倍时，效率较高。
- 经验数据表明，网格单元的大小取空间要素外包络矩形平均大小的3倍时，可极大的减少每个网格单元包含多个空间要素外包络矩形的可能性，获得较好的查询效率。

- 单元网格空间索引的效率在网格单元大小适中的时候非常高。
  - 由试验数据表明，空间要素个数在

200万左右时，如果网格单元大小划分合理的话,利用该索引查询进行过滤，响应时间最短可在2秒之内。

- 该类型索引适用于大型数据量的,空间范围确定的GIS应用。

❖ 5.3.1 KD树索引
❖ 5.3.2 KDB树索引
❖ 5.3.3 BSP树索引
❖ 5.3.4 G树索引

❖ 基于二叉树的索引技术
❖ 基于点的动态索引方法
❖ 1975年Bently提出的一种二叉树搜索树

❖ 1. KD树定义

- KD树的每个内部结点都包含一个

点，每个结点表示k维空间中的一个点，并且和一个矩形区域相对应，树的根结点和整个研究区域相对应。

- KD树要求用平行于坐标轴的纵横分界线将平面分为若干区域,使每个区域中的点数不超过给定值。
- 树中奇数层次上的点的x坐标和偶数层次上的点的y坐标把矩形区域分成两部分。
- 分界线仅起分界的作用,它的选取没有硬性的限制。一般选用通过某点的横向线或者纵向线。
- 分界线上的点,对左右分界线来说属于右部，对上下分界线来说属于上部。

定义

Kd树的每个节点表示K维空间的一个点，并且树的每一层都根据这层的分辨器（Discriminator）作出分枝决策。

Kd树的第i层的分辨器定义为：i MOD k（根节点为0层，下面依次加1）。

特征如下:

（1）左子树的所有节点的d维数值，均小于根节点的d维数值。

（2）右子树的所有节点的d维数值，均大于根节点的d维数值。

（3）左右子树也分别为kd树。

d为根节点的分辨器。

平面图

KD树的示意图

```
Algorithm KD_Search(R, P)
/*在根结点为R的KD树(子树)中查找点P。找
  到则返回结点，否则返回NULL*/
Begin
  If R=NULL Then Return NULL; //Not Found
  If R. Point=P Then Return R; //Found;
Else
  Begin
    d: =Discriminator of R;
    If P[d] <R Point[d] Then /*比较P点与
  R结点的第D维的值*/
      KD_Search(R.Lchild, P)//在左子树
  继续查找
    Else
      KD_Search(R.Rchild, P)//在右子树
  继续查找
  End
End.
```

```
Algorithm KD_Insert (R,P,F)
/*在根结点为R的KD树（子树）中插入点P，F为R的父结点
   */
Begin
  If R=NULL Then
    Begin
      Create a Node P;
      If F=NULL Then //KD树为空
        Root: =P//P成为根结点
      Else
        If R Is the Left child of F Then
          F.Lchild: =P//P作为F的左孩子
        Else F.Rchild: =P//P作为F的右孩子
    End
    Else
      Begin
        d: =Discriminator of R;
        If P[d]<R.Point[d] Then/*比较P点与R点的第D
   维的值*/
          KD_Insert (R.Lchild,P,R) //插入到左子树
   中
        Else
         KD_Insert (R.Rchild,P,R) // 插入到右子树
   中
      End;
End;


Algorithm KD_Delete (R,P)
```

```
/*在根结点为R的KD树（子树）中点P，删除成功返回True，否则
   返回False */
Begin
      Q: = KD_Search（R,P); //Q为要删除的对象
      LABEL;
      If Q=NULL Then Return False; //Not found
      If （Q.Lchild=NULL） And （Q.Rchild=NULL） Then
        Begin//第一种情况
          F: =Q is Father Node;
          If Q is the left Child of F then
            F.Lchild: =NULL
          Else F.Rchild: NULL;
            Delete Node Q;
          Return True;
       End
    Else
      Begin
         If （Q.Rchild=NULL） Then第三种情况转化为第二种
   情况处理
           Q.Rchild: =Q.Lchild;
        M: =FindMin（Q.Rchild);
      （Q）←（M); //将M结点的值赋给Q结点
       Q: =M; //让Q指向M结点
       GOTO LABEL; //继续删M结点
    End
end
```

- ❖特点
  - B-树向多维空间发展的一种形式
- ❖优点

- 较好的动态性
- 方便删除和增加空间点对象
- 无须周期性地调整索引自身结构

❖缺点

- 不直接支持占据一定空间范围的空间对象
  - 线、面

❖KDB树是KD树与B树的结合，它由两种基本的结构——区域页（region pages，非叶结点）和点页（point pages，叶结点）组成

- 点页存储点目标
- 区域页存储索引子空间的描述及指向下层页的指针。
  - 在KDB树中，区域页则显式地存储了这些子空间信息。区域页的子空间（如s11，S12和s13）两两不相交，

且一起构成该区域页的矩形索引空间（如S1）即父区域页的子空间。

图5-9 一颗KDB树的结构

- BSP树（Binary Space Partitioning Tree，二值空间划分树）是一种二叉树，它将空间逐级进行一分为二的划分。

- 优点
  - 很好地与空间数据库中空间对象的分布情况相适应
- 缺点
  - 深度较大，对各种操作不利

- G树是一种多层次的动态生长的格网结构。
  - 与KD树类似，按照循环交替的方式分割空间，但是它是采取平均分割空间的方法。
  - 假设各维的值，即有关的属性值，都能规范到0到1之间的值，并且每个区域中不能超过2点。如果超过2点，继续循环交替分割空间，直至每个区域不超过2点为止。
- 这种空间分割策略有3个特点
  - ①区域的二进制编码是全序的
  - ②分割所得的区域集合构成平面的一个划分
  - ③区域的二进制编码的位数越多，则该区域越小，它是其编码前缀所代表的区域的子空

间

- 5.4.1 R树索引
- 5.4.2 R+树索引
- 5.4.3 R*树索引
- 5.4.4 CELL树索引

- 1. R树及其特点
  - R树索引是一种高效的空间索引，它是B树在多维空间的扩展，也是平衡树。R树的结构类似于B+树的平衡树。
  - 对于一棵M阶的R树，R树中每个非叶子结点都由若干个（p，MBR）数据对组成。MBR（Minimal Boundary Rect）为包含其对应孩子的最小边界矩形。这个最小外接矩形是个广义上的概念，二维上是矩形，三维空间

上就是长方体MBV（Minimum Bounding Volume），以此类推到高维空间。p是指向其对应该子结点的指针。

- 叶子结点则是由若干个（OI，MBR）组成，其中MBR为包含对应的空间对象的最小外接矩形。OI是空间对象的标号,通过该标号可以得到对应空间对象的详细的信息。

❖ （a）空间实体分布　　（b）R树索引构造图

❖ 图5-15 R树索引数据结构示意图

❖ 2. R树查找

- *Algorithm R_Search* (N, W) {
- /*在根结点为N的R树中查找所有与W相交的数据矩形*/
- if (N.LEVEL==0) //N是叶子结点
- Return all data rectangles that intersect with W;
- else //N不是叶子结点
- for (i=1; i<N.COUNT; i++)
- if (N.MBRi; Intersect with W)
- *R_Search* (N.pi, W);
- }

❖ 3. R树插入

- *Algorithm R_Insert* (N, P) {
- /*向根结点为N的R树中插入数据矩形P*/
- if (N.LEVEL==0) {
- Insert P into N;
- if (N overfill) Split N;
- }
- else {//N是中间结点
- 〖Choose the entry in N whose rectangle needs least area enlargement to include the new data rectangle. Resolve ties

by choosing the entry with the rectangle of smallest area （Let's suppose it's entry is the answer);]

- *R_Insert* (N.pi, P);
- Adjust N.MBRi to enclose all rectangle in its child node;
- }
- }

❖ 4. R树删除

- *Algorithm R_Delete* (N, P) {
- /*从根结点为N的R树中删除数据矩形P*/
- if （N: LEVEL==0）
- {//N是叶结点
- if （N包含P）
- {
- 从N中删除P;
- N.COUNT=N.COUNT-1;
- return true;
- }
- else
- return false;
- }
- else
- {
- for （i =1; i<N.COUNT; i++）
- if （N.MBRi intersects with P）
- if （R_Delete (N.pi, P))
- if （N.pi, COUNT>=m）
- Adjust N.MBRi to enclose all child's rectangles;
- else

```
                    {
                        〖Reinsert all remain entries of
    N.pi and delete N.pi;  if N underfilled,  Reinsert all
                         remain entries of it and
                        delete it too...;〗
                    }
    }
}
```

❖

图5-17 一地图与其对应的R树结构

图5-18 图5-17地图中插入对象L后的R树结构

❖1. R+树及其特点

- R+树索引的主要特征是在R+树中兄弟节点对应的空间区域没有重叠,这样划分空间可以使空间搜索的效率提高。R+树也是R树的一个变种，在R+树中，兄弟节点对应的空间区域没有重叠,这样划分空间可以使空间搜索的效率提高。图5-20为一R+树对空间的划分及其索引对象的MBR组织。

图5-20 R+树索引示意图

- 2. R^+树查找
  - 算法*Search*（R，W）/*R: R+树的根结点，W: 查找矩形窗口*/
  - S1.[查找中间结点]
  - If R是非叶结点 then
  - For R的每一索引项（p，MBR） DO
  - If MBR∩W≠ϕ then Search（p，W∩MBR）
  - S2.[查找叶子结点]
  - If R是叶子结点 then
  - 检查R的每一数据项（OI，MBR）
  - RETURN所有与W相交的数据矩形
- **由查找算法可知，与R树相比，对于区域查找，查找路径应该可以减少，但依旧可能有多条；对于点查找，则可以通过一条路径得到查找结果。**

- **3. R^+树插入**
  - *Algorithm Insert*（R，IR）{
  - /*R为R+树的根结点，IR为要插入的数据矩形*/
  - I1.[查找中间结点]
  - if （R是非叶结点） then
  - for （p，MBR） do
  - if （MBR∩IR≠0） Insert（CHILD，IR);
  - I2.[查找叶子结点]
  - if （R是叶结点） then

- if （R已有M个数据项）then *SplitNode*（R);
- else 插入IR于R;
- }

❖ 4. R^+树删除

- *Algorithm Delete*（R，IR）{
- /*R为R+树的根结点，IR为要删除的数据矩形*/
- D1.[查找中间结点]
- if （R是非叶结点）then
- for R的每一索引项（p，MBR）do
- if （MBR∩IR≠0） then Delete(CHILD, IR);
- D2.[查找叶子结点]
- if （R是叶结点） then
- 从R中删除IR且调整R的父结点中对应的目录矩形;
- }

❖ 5. 结点分裂

Algorithm SplitNode（R）{
SN1[寻找一个划分]
调用Partition;
〖设（p，MBR）为与R相关联的索引项，S1与S2表示划分

```
    得到的两个子区域，创建两个新结点n1=（p1，MBR1）
    与n2=（p2，MBR2），MBRi=MBR∩Si，i=1，2;〗
SN2[填充新结点]
For （R的每一项（pk，MBRk） do
    if （MBRk∩MBR==MBRk） then // MBR k完全包含于
    MBRi
        put（pk，MBRk） in ni;
    else // MBR k与MBR1及MBR2都重叠。
        if （R是叶结点） then
            put （pk，MBRk） in n1 与n2;
        else
            〖用划分线继续分裂（pk，MBRk）所指结点，设
    得到的新结点为：nk1=
            （pk1，MBRk1），nk2=（pk2，MBRk2），MBRki完全
    包含于MBRi，将
            nki加入到ni，i=I，2;〗
SN3[向上传播结点分裂操作]
if （R是根结点）
    创建一新根结点，n1与n2为其两孩子结点;
else
    〖在R的父结点PR中，用n1与n2替换R。如果PR的索引项
    个数超过M，那么调用
    SplitNode（PR）。〗
}
```

- 针对R树和R+树在插入、删除与空间搜索效率两个方面难于兼顾的问题，产生

了CELL树索引。它在空间划分时不再采用矩形作为划分的基本单位，而是采用凸多边形来作为划分的基本单位，具体划分方法与BSP树有类似之处，子空间不再相互覆盖，如图5-21所示。CELL树的磁盘访问次数比R树和R+树少，由于磁盘访问次数是影响空间索引性能的关键指标，因此大大提高了搜索性能，故CELL树是比较优秀的空间索引方法。

❖ 图5-21 CELL树

❖5.5.1 点四叉树索引

- 5.5.2 MX四叉树索引
- 5.5.3 PR四叉树
- 5.5.4 CIF四叉树索引
- 5.5.5 基于固定网格划分的四叉树索引
- 5.5.6 线性可排序四叉树索引

- 点四叉树是QuadTree的一个变种，主要是针对空间点的存储表过与索引（Finkel and Bentley，1974），与KD树相似，两者的差别是在点四叉树中，空间被分割成四个矩形，四个不同的多边形对应于SW、NW、SE、NE四个象限。
- 对于k维数据空间而言，以新插入的点为中心将其对应索引空间分为两两不相交的2^k个子空间，依次与它的2^k个孩子结点相对应，对于位于某一子空间的点，则分配给对应的子树。
- 点四叉树的每个结点存储了一个空间点的信息及2^k个孩子结点的指针，且隐式地与一个索引空间相对应。其搜索过程和KD树相似，当一个点包含在搜索范围内时被记录下来，当一个子树和搜索范围有交叠时它将被穿过。如果想从

Point QuadTree中删除一个点的话，则会引起相应的子树的重建，一个简单的方法是将所有子树上的数据重新插入。图5-22是二维空间的一棵点四叉树的例子。

(a)平面图　　(b)结构图

图5-22 一颗二维的点四叉树结构

- 点四叉树的优点是结构简单，对于精确匹配的点查找性能较高。
- 其缺点有：①树的动态性差，删除结点处理复杂；②树的结构由点的插入顺序决定，难以保证树深度的平衡；③区域

查找性能较差；④对于非点状空间目标，必须采用目标近似与空间映射技术，效率较差；⑤不利于树的外存存储与页面调度；⑤每个结点须存储2k个指针域且其中叶子结点中包含许多空指针，尤其是当k较大时，空间存储开销大，空间利用率低。

- MX四叉树索引即Matrix四叉树索引。在k维空间中，整个数据空间被分割成四个矩形。四个不同的多边形对应于SW、NW、SE、NE四个象限。每次分割空间时，都是将一个正方形分成四个相等的子正方形，依次重复地进行2^k次等分，直到每个正方形的内容不超过所给定的桶量（比如一个对象）为止，空间中的每一点都属于某一象限且位于该象限内，每一象限均只与一个空间相关联。
- 在MX四叉树中，叶子结点的黑结点或空结点分别表示数据空间某一位置空间

点的存在与否。图5-22为二维空间的一棵MX四叉树的例子。

图5-23 一颗二维的MX四叉树例子

- PR四叉树是点四叉树的一个变种，它不使用数据集中的点来分割空间。在PR四叉树中，每次分割空间时，都是将一个正方形分成四个相等的子正方形，依次进行，直到每个正方形的内容不超过所给定的桶量（比如一个对象）为止。
- PR四叉树与MX四叉树的主要区别是：①

叶子结点可能不在树的同一层次；②PR四叉树的叶结点数及树的深度都小于MX四叉树，因此PR四叉树的检索效率要高于MX四叉树。

图5-24 PR四叉树的索引结构

- CIF (Caltech Intermediate From) 四叉树是针对表示VLSI (Very Large Scale Integration) 应用中的小矩形而提出的，它可以用于索引矩形及其他形体。

- 它的组织方式与区域四叉树相似，数据空间被递归地细分直至产生的子象限不再包含任何矩形。在分解的过程中，与任一划分线相交的矩形与该划分线对应的象限相关联，属于一个象限的矩形不能属于祖先象限，换句话说，矩形只属于完全包围它的最小象限。图5-25是二维空间一颗CIF树的例子（这里假设数据桶的容量为3个矩形）。

图5-25 二维空间CIF四叉树的一个例子

- 在基于固定网格空间划分的四叉树空间索引机制中，二维空间范围被划分为一系列大小相等的棋盘状矩形，即将地理空间的长和宽在X和Y方向上进行2^N等分，形成$2^N \times 2^N$的网格，并以此建立N级四叉树。
- 在四叉树中，空间要素标识记录在其外包络矩形所覆盖的每一个叶结点中。
  - 但当同一父亲的四个兄弟结点都要记录该空间要素标识时，则只将该空间要素标识记录在该父亲结点上，并按这一规则向上层推进。

  - 该方法与传统四叉树索引的不同之处有两点：
    - 四叉树结点编码方式不同，
    - 结点和空间要素的对应关系不同。
  - 编码方式 ：

- 将四叉树分解为二叉树，即在父结点层与子结点层之间插入一层虚结点，虚结点不用来记录空间要素;
- 按照中序遍历树的顺序对结点进行编码，包括加入的虚结点。

- 空间查询时:
  - 根据查询区域生成所要搜索的结点编号的集合;
  - 由于新的编码方式，孩子和父亲结合编号的连续性将结点编号集合变换成连续的结点编号的范围，这样就可以很容易的用SQL模型构造条件,从索引表中检索出满足要求的空间要素。

❖5.6.1 网格文件

❖5.6.2 R文件

- 网格文件的基本思想是根据一个正交的网格（orthogonal grid）划分k维的数据空间，如图5-28所示。
- 网格是用k（数据的维数）个一维的数组来表示的，这些数组称为刻度（scales)。刻度的每一边界

（boundary）构成一个（k-1）维的超平面（hyper plane），对于二维空间为平行于x或y轴的直线，这一超平面将数据空间划分为两个子空间。所有的边界一起将整个数据空间划分成许多k维的矩形子空间，这些矩形子空间称为网格目录（grid directory），由一个k维的数组表示。

- 目录项（即网格目录数组的元素）和网格单元（grid cells）之间具有一对一的关系。网格目录的每一网格单元包含一个外存页的地址，对应着一个数据桶，一般一个数据桶为硬盘上一个磁盘页，这一外存页存储了包含了网格单元的数据目标，称为数据页（data page）。数据页所对应的一个或多个网格单元称之为存储区域（storage region），存储区域两两不相交。每个数据桶往往可以包含着几个相邻的单元，存储多个网格单元的目标，只要这几个网格单元一起形成一矩形的区域。

❖

❖

❖ 图5-28 网格文件的结构

- 空间填充曲线是一种重要的近似表示方法，将数据空间划分成大小相同的网格，再根据一定的方法将这些网格编码，每个格指定一个唯一的编码，并在一定程度上保持空间邻近性，即相邻的网格的标号也相邻，一个空间对象由一组网格组成。这样可以将多维的空间数据降维表示到一维空间当中。
- 普通的关系数据无法对多维数据直接进行查询，通过使用空间填充曲线对空间实体数据集进行降维处理，映射到一维空间进行编码，就可以重复利用已有的B树索引、Hash索引、Bitmap索引等技术针对一维空间进行查询。
- 理想的空间映射方法是：在多维空间中聚集的空间实体，经过填充曲线编码以后，在一维空间中仍然是聚集的。

（a）行排序　　（b）Hilbert排序　　（c）Z排序

图5-30 几种常用的空间填充编码方法

- Z-排序（Z-ordering）技术将数据空间循环分解到更小的子空间（被称为Peano Cell），每个子空间根据分解步骤依次得到一组数字，称为该子空间的Z-排序值。子空间有不同的大小，Z-排序有不同的长度，显然，子空间越大，相应的Z-排序值越短。这里，分辨率（resolution）是指最大的

分解层次，它决定了Z-排序值的最大长度。

图5-31 Z-排序示例

- 假设一个点的座标为X=011，Y=101，该点的线性编号Z-order值计算过程为：
- 取X的第3位、第2位、第1位的二进制值分别作为Z-order值的第6位、第4位和第2位的二进制值；再取Y的第3位、第2位、第1位的二进制值分别作为Z-order值的第5位、第3位和第1位的二进制值，最终结果为011011（十进制为27）。

- 与Z-排序类似，Hilbert曲线也是一种空间填充曲线，它利用一个线性序列来填充空间，其构造过程如图5-33所示。
- 理想情况下，这种映射会带来更少的磁盘访问，但由于磁盘访问的次数依赖于很多因素，如磁盘页面容量、分割算法、插入顺序等，因此，对于不同的查询，其磁盘访问的次数会有很大的不同。通常，可将给定查询代表的子空间中每个网格点的散列单元平均数，来作为衡量磁盘访问效率的标准。
- 实验证明，Hilbert曲线的方法比Z-排序好一些，因为它没有斜线。不过Hilbert曲线算法的计算量要比Z-排序复杂。

图5-33 Hilbert曲线示例

- Hilbert曲线的算法如下（Faloutsos and Roseman，1989）:
  - （1）读入x和y坐标的n比特二进制表示。
  - （2）隔行扫描二进制比特到一个字符串。
  - （3）将字符串自左至右分成2比特长的串si，其中i=1，...，n
  - （4）规定每个2比特长的串的十进制值di，例如“00”等于0，“01”等于1，“10”等于3，“11”等于2。
  - （5）对于数组中每个数字j，如果j=0把后面数组中出现的所有1变成3，并把所有出现的3变成1。j=3把后面数组中出现的所有0变成2，并把所有出现的2变成0。
  - （6）将数组中每个值按步骤5转换成二进制表示（2比特长的串），自左至右连接所有的串，并计算其十进制值。

图5-34 Hilbert曲线转换例子

空间数据查询、访问

- SQL查询语言概述
  - 1974年由Boyce和Chamberlin提出，在IBM公司San Jose实验室研制的System R上实现了这种语言
  - 1986年成为ANSI标准
  - 1987年成为ISO标准
- 优点
  - 非过程化语言

- 只需指明“做什么”，而不必指明“怎么做”，无需了解存取路径及操作过程。
- 统一的语言
  - 语言功能极强但十分简洁，只用了9个动词；而且语法简单，接近口语，易学易用
- 所有关系数据库的公共语言

❖SQL查询语言的功能
- DDL数据定义语言
- DML数据操纵语言
- DCL数据控制语言

❖数据查询是GIS的一个重要功能，一般定义为作用在GIS数据上的函数，它返回满足条件的内容。查询是用户与系统交流的途径。
❖查询是GIS用户最经常使用的功

能，用户提出的很大一部分问题都可以以查询的方式解决，查询的方法和查询的范围在很大程度上决定了GIS的应用程度和应用水平。

- ❖空间数据库查询语言是指从在空间数据库中查找出所有满足空间约束条件和属性约束条件的地理实体的算法语言。
- ➢常规的关系数据库查询语言是SQL，它可以作为属性数据库的查询语言。如:

  Select * from city ,province where provice.name=“江苏省”

- 关系模型的扩展
  - 突破关系模型中关系必须是第一范式的限制,允许定义层次关系和嵌套关系
  - 增加抽象数据类型
  - 增加空间谓词
  - 增加适合于空间数据索引的方法

- 图形数据的存储和管理可以由一个扩展关系DBMS来实现
  - 用统一的DBMS来管理图形和属性数据
  - 图形数据管理也可以享用DBMS在数据管理方面带来的优越性
  - 图形数据的关系化表达,使其能享用客户机/服务器优势

- 扩展SQL以处理空间数据（基于空

间的对象模型）

- 针对所有几何类型的基本操作
- 描述空空间对象间拓扑关系的函数
- 空间分析的一般操作

❖OGIS标准的局限性

- 局限于对象模型
- 对于简单的选择-投影-连接查询，操作有局限性
- 过于关注基本拓扑的和空间度量的关系，而忽略了对整个度量操作的类的支持
- 不支持动态的、基于形状以及基于可见性的操作

❖SQL3/SQL99

- 对SQL进行对象-关系扩展的一个标准平台
- 不是专门针对GIS或者空间数据库

- SQL3是SQL的最新标准
  - 对SQL语法规则作出了更加详细和准确的定义
  - 对空间数据的支持也作出了统一的描述
    - 空间数据类型在数据库中的存储
    - 定义操作与空间数据的空间运算符

- 点线面定义
  - 0维点Points
  - 1维环Planar、1维曲线Curves
    - ST_LineString
    - ST_CircularString
    - ST_CompoundString
  - 2维面Surface
    - ST_Polygon
    - ST_CurvePolyon
- 空间对象定义
  - ST_Geometry: 表示多个元素形成的集合对象

- 抽象数据类型（ADT）
  - 指一个数学模型以及定义在该模型上的一组操作。
  - ADT包括数据数据元素，数据关系以及相关的操作。

    即ADT

    {

    数据对象:（数据元素集合）

    数据关系:（数据关系二元组结合）

    基本操作:（操作函数的罗列）

    }
  - 在面向对象编程语言中，像C++、Java都能较好的支持ADT，如类的机制。而在C语言中缺少了对相关方法的支持。
  - 抽象数据类型需要通过固有数据类型(高级编程语言中已实现的数据类型)来实现。抽象数据类型是与表示无关的数据类型，是一个数据模型及定义在该模型上的一组运算。对一个抽象数据类型进行定义时，必须给出它的名字及各运算的运算符名，即函数名，并且规定这些函数的参数性质。一旦定义了一个抽象数据类型及具体实现，程序设计中就可以像使用基本数据类型那样，十分方便地使用抽象数据类型。

❖行类型

- 用于定义关系的类型,指定了关系的模式

❖ SQL扩展方法

- 将空间运算结果作为新的属性值添加于原有关系笛卡尔积的末尾。
- 这些派生属性与其他原始属性一样可以应用于SELECT、WHERE或HAVING语句中。
- 由于空间算子无法直接放置于FROM语句中,所以需使用嵌套的SQL语句。
  - 例如,下面查询中的**FROM**语句就使用了这样的子查询。

```
SELECT lu.ID, sl.ID,
lLocation, areaval
FROM
(SELECT *,
INTERSECTION(lu.Location,
sl.Location) AS lLocation,
AREA(lLocation) AS areaval
FROM landuse AS lu,
soil AS sl)
WHERE lu.type=
```

‘Brushland’ and sl.type= ‘A’ and areaval>700 and areaval<900

<FROM子句>::=<FROM><关系表>[,<子查询>]
<关系表>::=<表名>[{,<表名>...}]
<子查询>::=SELECT<子SELECT语句>FROM<空间关系表>
<子SELECT语句>::=*,<派生属性>
<派生属性>::=<空间操作算子>AS<属性名>{,<派生属性>}
<空间操作算子>::=<一元空间操作算子>|<二元空间操作算子>
<二元空间操作算子>::=<几何算子>|<拓扑关系算子>|<创建算子>

- 空间操作算子是指带有参数的函数
  - 以空间特征为参数，返回空间特征或数值。
  - 空间操作算子主要分为两类
    - 一元空间操作算子

      XC(Pnt)、YC(Pnt)

SP(Poly/ARC)、EP(Poly/ARC)
ARCS(Pgn)
CENTROID(Pgn)
LENGTH(Poly/Pgn)
AREA(Pgn)
VORONOI(Pnt)
BUFFER(Location)
FUSION(Location)

- 二元空间操作算子
  - 二元几何算子
  - 二元拓扑关系算子
  - 二元创建算子

❖ 1. 二元几何算子
- 主要指距离算子
- **DISTANCE(Pnt, Pnt)** 和**DISTANCE(Pnt, Poly)**

❖ 2. 二元拓扑关系算子
- 判断两空间特征之间基本拓扑关系的算子
  - **DISJOINT(Location，Location)** 分离关系

- **OVERLAP(Location，Location)** 相交关系
- **NEIGHBOUR(Location，Location)** 相邻关系
- **TOUCH(Location，Location)** 相触关系
- **CONTAIN(Location，Location)** 包含关系
- **EQUAL(Location，Location)** 相等关系

❖ 3. 二元创建算子

- 从已有的两个空间特征通过运算得到新的特征及属性的算子
  - **UNION(Pgn,Pgn)** 并叠置
  - **DIFFERENCE(Pgn,Pgn)** 差叠置
  - **INTERSECTION(Pgn,Pgn)** 交叠置

❖实例1

- 建立一个试验室，选址标准
  - 土地利用类型为矮灌木地
  - 土壤类型为“A”
  - 地址必须位于现存下水道300米范

围内

- 地址须位于现存溪流20m以外
- 地址的面积须大于2000m^2

```
SELECT labLocation
FROM (SELECT *,
INTERSECTION(lu.Location, sl.Location)
AS lsLocation,BUFFER(sw.Location, 300)
AS buflLocation,
INTERSECTION(lsLocation, buflLocation)
AS lsbLocation,
BUFFER(sm.Location, 20) AS
buf2Location,
DIFFERENCE(lsbLocation, buf2Location)
AS labLocation,
AREA(labLocation) AS areaval
      FROM landuse AS lu, soil AS sl, sewer
AS sw, stream AS sm)
WHERE lu.type='brushland' and
sl.type='A' ad areaval＞2000
```

❖实例2

- 对土地作土地适宜性评价,即要求出所有地块的适宜程度

- 叠加土地利用与土壤图
- 对叠加后的地块进行评价
- 合并具有相同适宜性等级、相邻且面积大于100㎡的地块

```
SELECT FUSION(ILocation)
FROM(SELECT *, INTERSECTION(lu.Location,
    sl.Location) AS ILocation,AREA(ILocation)
    AS areaval
                    classfyval=(CASE lu.type‖
    sl.type
                WHEN 'BrushlandA' THEN 'Ⅲ'
                WHEN 'BrushlandB' THEN 'Ⅱ'
                WHEN 'WaterA' THEN 'Ⅰ'
                WHEN 'WaterB' THEN 'Ⅱ'
                WHEN 'ForestA' THEN 'Ⅱ'
                WHEN 'ForestB' THEN 'Ⅱ'
                END)
          FROM landuse AS lu, soil AS sl)
WHERE ILocation <> NULL and areaval>100
        GROUP BY classfyval
```

空间关系查询

- 空间选择查询也称为范围查询
  - 在地图上划出一个区域(称为查询区

域，查询该区域内的所有空间数据
  - 空间查询的基础

❖ 主要空间查询包括
  - 点查询
  - 区域查询
  - 最邻近查询

❖ 点查询
  - 根据用户在屏幕上选择的点的屏幕坐标判断与它临近距离小于给定阈值的地图单元,检索出所有包含该点的空间对象，并且用特殊方式显示

❖ 区域查询
  - 通过屏幕上设定一个矩形框,检索出所有在空间中与该区域相交的空间对象
  - 窗口或开窗查询时一种特殊的区域查询
  - 采用R+空间索引技术

- 最邻近查询
  - 查找一个距离某个给定查询点最近的空间对象

- 由于空间数据的复杂性，空间查询效率成为空间数据库性能的瓶颈

- 1 减小候选集
- 过滤筛选过程的对象近似技术
  - 简单，实现快速的过滤
  - 为有效降低计算量,空间索引大都采用空间实体进行近似表示的策略
    - 最小边界圆MBC
    - 最小边界矩形MBR
    - 最小边界m边形MBM

- 与MBR相比，其它近似描述的空间存取方法需访问更多的数据页，但由于它们的近似质量高，可减少未命中对象，从而减少后继处理中的几何计算开销
- 近似使用的参数越多，近似质量越高，要在近似性能和额外存储之间达到较佳折中

- 2 提高几何检测速度
- 空间查询精炼步骤中的相关技术
  - 平面扫描技术
    - 将多边形对象分解成最小数量的互不相交的不规则多边形
    - 计算几何中常用的算法
    - 快速判断两个多边形是否相交
  - 对象分解技术

- 将对象分解为满足一些定量约束的简单子对象
- 常用的精确几何处理技术
- 非常适合于空间查询

❖空间连接查询是空间数据库系统一种重要的多路查询，即从两个数据集合众检索出所有满足某一条件的空间对象。

- 例如:判断两张地图上的空间对象是否相交

❖1 空间连接的分步骤处理过程

- 过滤
  - 产生满足空间谓词的所有最小边界矩形（MBR）对，这些MBR对是下一步操作的候选者，其中可能包含一些不满足空间谓词的结果
- 精炼

- 检查每一对候选者，根据它们相应空间对象的确切范围选出满足空间谓词的结果。此步骤需要检查候选集中每个元素的精确几何信息和精确的空间谓词。

❖MBR空间连接方法

- 中心思想：如果两个对象的MBR（最小包围矩形）不相交，它们相对应的对象也不相交。

❖2 嵌套循环连接方法

- 基本思想
  - 产生数据集R和S所有可能的对象偶
- 方法
  - 简单嵌套循环连接方法
  - 嵌套块循环连接方法
  - 有索引的嵌套循环方法

- 空间连接查询
  - 基于空间索引的空间连接查询
    - 基于同步R树遍历空间连接算法
    - 种子树连接方法
  - 基于其它索引的空间连接查询
    - 分区空间合并连接
    - 空间哈希连接

- 基于同步R树遍历空间连接算法

- 种子树连接方法

- 分区空间合并连接
  - 算法
    - 计算所有分区
    - 对每个对象集扫描
    - 连接所有对应的分区对
    - 排序结果并消除重复现象

- 空间哈希连接
  - 算法:
    - 计算分区数目
    - 取样对象集A，并初始化分区
    - 扫描对象集A，将对象插入分区边界
    - 扫描对象集B，将B插入前一步划定的分区，并在必要时重复存放对象
    - 连接所有对应分区中的对象对，并返回结果

- 由于空间数据的复杂性，空间查询效率成为空间数据库性能的瓶颈
- 查询优化技术的研究是空间数据库应用的难点和突破点
- 查询效率的提高主要来自于外部环境和应用程序
- 优化策略
  - 空间索引技术

- 查询路径优化
- 数据压缩以及缓存

❖查询优化器

- 数据库软件中的一个模块,用于产生不同的计算计划并确定适当的执行策略
- 承担任务
  - 逻辑转换
  - 动态规划

❖执行查询分析的任务

- 按照经历分析后得到的语法树,通过调用客户端提供的各种查询接口来获得结果,最后输出用户期望的结果

❖查询分析的类型

- 属性查询
- 空间查询
- 空间分析

时态空间数据库

- 主要内容
  - 地理信息的时空分析
  - 时态数据库
  - 时空数据模型
  - 时空数据库

- 空间数据库是对与地理空间相关的数据进行有效管理的系统。然而，现实世界的数据不仅与空间相关，而且与时间相关。
- 现有空间数据库大多不具有处理数据的时间动态性的功能，而只是描述数据的一个瞬态（snapshot ）。无法对数据变化的历史进行分析，更无法预测未

来的趋势。亦称为静态空间数据库。

- ❖ 能提供完善的时序分析功能，高效地回答与时间相关的各类问题，有效地处理与时间相关的空间信息，即所谓时态空间数据库。

- ❖ 地理世界的时间
  - 1. 时间概念
  - 2. 时间的结构
  - 3. 时间的表示
  - 4. 时间粒度
  - 5. 时间的密度特性
  - 6. 时间按的不确定性
  - 7. 事件与状态

- ❖1 时间的概念

❖ 2 时间的结构

- 线性结构
  - 认为时间是一条没有端点，向过去和将来无限延伸的线轴，除了与空间一样具有通用性、连续性和可测量性外，还具有运动的不可逆性（或称单向性）和全序性。
- 循环结构
  - 反映了时间的周期性、稳定性，与时间的线性结构不可分割，相辅相成，形成了现实世界在继承中的发展。
- 分支结构
  - 分为单向分支结构和双向分支结构，分别反映了具有不同的历史时间结构和未来时间结构的多个目标现象的时间结构，其中各分支具有两两正交性。
- 多维结构
  - 是同一目标的演变经历，从不同时间角度来看，体现为时间的多维结构。

❖ 多维结构主要表现

- （1）有效时间（Valid Time）Tv:

空间目标从产生到消亡，是它在现实世界中存在的时间区间，称为有效时间(也称世界时间、数据时间、逻辑时间、事件时间)。如果理论模型允许目标消亡后再生，则有效时间是多个不相交的时间区间的并。

- （2）数据库时间（Database Time）Td: 目标数据输入系统的时间，称为数据库时间（或事务时间、物理时间、执行时间、系统时间）。
- （3）用户定义时间(User-defined Time): 用户根据需要自己为目标标注的时间，称为用户定义时间，只有用户知道其语义，DBMS不能解释，语义由具体应用确定。
- 另外还有决策时间、观察时间等等。

❖3 时间的表示

❖4 时间粒度（多标度性）

- 指用于度量时间的尺度的多样性,时间标度也称时间分辨率或时间粒度。
- 不同的应用领域,及同一应用领域中的不同应用范围,都可能采用不同的时间标度。
- 时间标度的选择存在着理想的时间精度和节约内存开销相互权衡的问题。

❖5 时间的密度特性

- 时间的密度特性体现为以下模型:
  - 离散模型：时间与自然数同构，每个自然数对应一个时间粒子，是一种较常用的结构;
  - 紧凑模型：时间与有理数/实数同构;
  - 连续模型：时间与实数同构，每个实数对应时间上一个点。

❖6 时间的不确定性

- 空间数据库中的数据在空间、非空间属性上都具有不确定性，同样在时态性上也存在着不确定性。
- 当某事件发生是已知的，但何时发生是未知的，则称该事件是时态非确定的。

❖7 事件与状态

❖时态数据库定义

- 存储现实世界的时间经历状态信息的数据

❖传统数据库只反映了一个对象的发展全过程中在某一个时刻的状

态，不能很好的反映、存储其过去和未来

- 管理时态信息的要求
  - 要求管理被处理时间的历史性信息
  - 要求管理数据库系统中元时间的时态信息

- 非空间的时态数据库的研究重点
  - 时态数据模型
  - 时态查询语言
  - 时态存储结构
  - 时态实现技术

- 时态数据库的基本概念
  - 元组时刻标记
  - 元组时区标记
  - 元组时态元素标记
  - 属性时刻标记
  - 属性时区标记

- 属性时态元素标记

❖ 时态数据库的基本类型

- 根据数据库处理时间的能力分类
  - 静态数据库
    - 反映现实世界某一瞬间情况的数据模型。它记录了特定时刻的数据库状态。快照数据库采用这样的假设：一个存储在数据库中的元组，一定是真实世界中的有效事实。
  - 回滚式数据库
    - 数据库中被管理对象的生命周期是事务时间的数据库。它保存了数据库中事务提交、状态演变的历史状态。
  - 历史数据库
    - 数据库中被管理对象的生命周期是对象的有效时间，每一个元组记录了数据的一个“历史”状态。历史数据库中没有约束时间的表示方法，可以是时间点的集合、时间区间或者区间集合等形式表示。
  - 双时序数据库
    - 数据库中元组包含一个系统支持的有效时间和一个系统支持的事务时间的数据库，称为双时态数据库。双时态数据库具备了快照数据库、历史数据库和

回滚数据库的特点，存储了现实世界和数据库系统的变更历史。

- 根据数据库存放的内容分类
  - 历史数据库
  - 实时数据库
  - 预测数据库
- 根据数据库中对象的时间结构分类
  - 线性数据库
  - 分支数据库
  - 周期数据库
- 根据对象的状态和时间分类
  - 基于状态的数据库
  - 基于时间的数据库
- 时空模型分类
  - 基于位置的时空数据表达

- 快照模型，网格模型
- 基于空间实体的时空数据表达
  - 矢量修正模型，面向对象的时空模型
- 基于时间的时空数据表达
  - Peuqent和Duan提出的基于时间的表达模型
- 时空的综合表达方法
  - 时空复合模型

❖ 此模型在快照数据库（Snapshot Database）中仅记录当前数据状态，数据更新后，旧数据的变化值不再保留，即“忘记”了过去的状态。

❖ 序列快照模型是将一系列时间片段快照保存起来，反映整个空间特征的状态，根据需要对指定时

间片段的现实片段进行播放。

- ❖ 缺点
  - 由于快照将未发生变化的时间片段的所有特征重复进行存储,会产生大量的数据冗余，当应用模型变化频繁，且数据量较大时，系统效率急剧下降。此外，此模型不表达单一的时空对象,较难处理时空对象间的时态关系。

- ❖ 避免了序列快照模型的数据冗余问题

- ❖ 为了避免连续快照模型将每张未发生变化部分的快照特征重复进行记录，基态修正模型按事先设定的时间间隔采样，不存储研究区域中每个状态的全部信息，只存贮某个时间的数据状态(称为基态)，以及相对于基态的变化量。

基态修正的每个对象只需存储一次，每变化一次，只有很小的数据量需要记录；同时，只有在事件发生或对象发生变化时才存入系统中，时态分辨率值与事件发生的时刻完全对应。基态修正模型不存储每个对象不同时间段的所有信息，只记录一个数据基态和相对于基态的变化值，提高了时态分辨率，减少了数据冗余量。毫无疑问，在基态修正法中，检索最频繁的状态作为基态(一般的用户最关注的是“现在”时 ，即系统最后一次更新的数据状态)。此外，目标在空间和时空上的内在联系反映不直接，会给时空分析带来困难。

- 时空复合模型将空间分隔成具有相同时空过程的最大的公共时空单元，每次时空对象的变化都将在整个空间内产生一个新的对

象。对象把在整个空间内的变化部分作为它的空间属性，变化部分的历史作为它的时态属性。时空单元中的时空过程可用关系表来表达，若时空单元分裂时，用新增的元组来反映新增的空间单元，时空过程每变化一次，采用关系表中新增一列的时间段来表达，从而达到用静态的属性表表达动态的时空变化过程的目的；但在数据库中对象标识符的修改比较复杂，涉及的关系链层次很多，必须对标识符逐一进行回退修改。

- 时空立方体模型用几何立体图形表示二维图形沿时间维发展变化

的过程，表达了现实世界平面位置随时间的演变，将时间标记在空间坐标点上。给定一个时间位置值，就可以从三维立方体中获得相应截面的状态，也可扩展表达三维空间沿时间变化的过程。缺点是随着数据量的增大，对立方体的操作会变的越来越复杂，以至于最终变的无法处理。

- 时空数据库（Spatio-temporal Database，STDB）
  - 时态数据和空间数据的结合,即对空间数据库赋予时态特征
- 时空数据库系统或数据处理技术

的主要内容

- 空间时态数据的表达
- 空间时态数据的更新
- 空间时态数据的查询

❖ 时空数据库实现的两种途径

- 扩展传统的关系模型
- 采用面向对象方法

❖ 由于传统关系模型语义丰富、理论完善以及具有许多高效灵活的实现机制，使人们开始尝试在传统关系模型中加入时间维，扩充关系模型，用关系代数及查询语言来处理时态数据，从而直接或间接地基于关系模型支持时空数据的存贮、表示和处理。

❖ 基于这一思想，主要有下列方法。

- 1. 归档保存
- 2. 时间片
- 3. 记录级时间戳

❖1. 归档保存

❖这是一种支持时态数据的最原始、最简单的方法。就是以规则的时间间隔备份所有存贮在库中的数据。

❖缺点

- （1)发生在备份间的事件未被记录，致使部分信息丢失;
- （2）对存档信息的搜索慢且笨拙;
- （3）许多数据重复归档，存在大量的数据冗余。

❖2. 时间片

❖这种方法是将库中某时刻的时空信息存贮在一个平面文件或二维

表格中，即所谓时间片（time-slice）。当发生变化时，将当前状态表存贮起来，并给定一个时间戳（用Since和until来标记，表明状态的一段区间），然后复制一个并更新为新状态。与归档保存相比，这一方法在效率上有所改善。但仍存在大量的数据冗余，而且当使用一个时间戳时，对有关生命周期的查询会非常繁琐。采用两个时间戳时，对特定属性变化的查询，又需检测所有时间片。

- 3. 记录级时间戳
- 这种方法将时间戳作用于记录（或元组）级，而非整个关系，时间戳可采用前面提到的两种方法。其实现过程是，

当发生事件时，将当前记录标记时间戳，然后建立一个具有变化后新属性值的记录加入表中。新记录的加入可以有三种不同的方法。最简单低效的方法是把新记录加在表尾。这将得到一个规整的时序视图，但也意味着需要频繁地顺序搜索来应答查询。

- Snodgrass和Aln描述了另一种方法，将相关的记录依时序放在一起。这种方法对于关于生命期的查询非常便利。第三种方法是对每一时间片以同样的方式对表中记录排序。这种方法的问题在于若发生一事件时，某记录没有发生变化，那它仍需复制或以空白填充不变的记录。

- 所有这些上述方法存在一个共同缺陷：关系表会变得越来越长，导致应答时间的下降。
- Lum等[68]提出了另一种称作链

式元组级时态GIS实现方法。工作原理是由两个关系而不是一个关系来表示时态实体。第一个关系只存贮当前状态，每当事件发生时被更新。第二关系以链式保存所有历史记录。这样在相关的记录间建立了简单的遍历存取路径，提高了效率，而且删除记录也非常容易，但需要整个记录时并不方便。一种改进的方法是分离时变属性与非时变属性，从而节省了内存开销，对历史数据存取快速，减少了更新费用。Iarine等1993年利用类似的原理处理了空间属性的变化历史。

- 关系模型的数据类型简单，缺少

表达能力，GIS中的许多实体和结构很难映射到关系模型中。

- 以更自然的方式表示复杂的地理信息。其中OO方法已引起时态GIS设计者的很大兴趣。

- 对复杂的时间信息，当今大部分基于关系模型的GIS是通过大量元组牵强地表示，对一些无法表示的语义属性只能在外部描述。而在OO模型中，提供了广泛化、特例化、聚合和关联等机制，易于支持时态GIS中各种形式的时空数据，其中可以使用矢量数据或栅格数据，也可以是不同数据类型的集成。数据结构和方法的封装便于数据对象不同表示间的

转换。在处理时空不确定性方面，OO技术也体现了优越性。

- OO方法已逐渐被时态GIS采用。Mncler1993年提出了把时空图集合看作一个时态图集对象，体现了高效、方便的优点。Beller等1991年也在他们的时态GIS中实现了类似的方法。
- 在面向对象的时态GIS研究中，较为典型的成果有ZMith OO模型和DSAM*/T模型。

- Zmith OO模型[69]提供了唯一的对象标识，将对象完全封装起来，用灵活的相关语义说明内部对象的关联。版本化的实现是通过使用has-version关系，将当前状态

中的对象与过去不同时刻的对象状态相关联，每个版本又使用Predecessor/Suclessor关系与其前后的版本相连接。这一机制方便了对象的版本集合或某个版本的存取。时态维的实现是通过在对象结构的适当层次上附加时间成为的方式，可在线性版本序列或版本树中描述时态拓扑关系。

- OSAM*/T模型[70]使用了对象时间戳方法，记录对象、对象实例的历史和对象间关联的历史，使历史数据和当前数据在物理上、逻辑上分离，历史区可采用分布式存贮或静态存贮。该模型的不

足之处在于未能对物理时间给予支持。

- Monia和Richard提出的一个基本OO语义的时空数据模型通过对象和事件在数据模型中相互作用的方法，集成了空间和时间维。

空间数据元数据与空间数据共享

- 主要内容
  - 空间数据元数据
  - 空间数据共享

- 空间数据元数据概念与分类
- 空间数据元数据的主要内容
- 空间数据元数据的主要作用及功

能

- 空间数据元数据的标准及实例
- 空间数据元数据组织和管理

- 元数据（Metadata）
  - Meta：希腊语词根，意思是“改变”
  - Metadata：原意是关于数据变化的描述
  - 一般认为元数据就是“关于数据的数据”，即关于数据的描述性数据信息
  - 在地理空间数据中，元数据是说明数据内容、质量、状况和其他有关特征的背景信息。

- 随着计算机技术和GIS技术发展，特别是网络通信技术的发展，空间数据共享日益普遍。

- 管理和访问大型数据集的复杂性正成为数据生产者和用户面临的突出问题。
- 在这种情况下，空间数据的内容、质量、状况等元数据信息变得更加重要，成为信息资源有效管理和应用的重要手段。
- 地理信息元数据标准和操作工具已经成为国家空间数据基础设施的一个重要组成部分。

- 元数据的概念
  - 元数据是关于数据的描述性数据信息,它应尽可能多地反映数据集自身的特征规律,以便于用户对数据集的准确、高效与充分的开发与利用，不同领域的数据库,其元数据的内容会有很大差异。

- 通过元数据可以检索、访问数据库，可以有效利用计算机的系统资源,可以对数据进行加工处理和二次开发等。

❖到目前为止，科学界关于元数据认识的共同点是：元数据的目的就是促进数据集的高效利用，并为计算机辅助软件工程（CASE）服务。

❖元数据与数据字典的区别

❖空间数据源数据的分类

- 数据集系列Metadata
- 数据集Metadata
- 要素类型和要素实例Metadata
- 属性类型和属性实例Metadata

❖元数据的内容包括:

1）对数据集的描述；对数据集中各数据项、数据来源、数据所有者及数据序代（数据生产历史）等的说明;

2）对数据质量的描述，如数据精度、数据的逻辑一致性、数据完整性、分辨率、元数据的比例尺等;

3）对数据处理信息的说明，如量纲的转换等;

4）对数据转换方法的描述;

5）对数据库的更新、集成等的说明。

❖描述空间信息的空间元数据内容按照部分、复合元素和数据元素来组织

- 1、 部分
- Metadata部分是定义具有相互关联的Metadata数据元素以及复合元素集合的一个Metadata子集。
- 2、 复合元素
- 复合元素由一组数据元素和其他复合元素组成,它是代表高层次的不能用单个数据元素描述的概念。
- 3、 数据元素
- 数据元素是数据的最小组成单位,即图元Metadata，它由数据元素的名称、定义、约束条件、最大次数、数据类型、域值等组成。

❖空间数据标准体系的内容分为8个基本内容和4个引用部分，共12部分组成

- 第一层：目录层
  - 提供空间元数据复合元素和数据元

素是数字地球中查询空间信息的目录信息

- 第二层：空间元数据标准的主体
  - 包括了全面描述地理空间信息的必选项、条件可选项以及可选项的内容

❖ 元数据对数据生产者和用户的作用

- 1）帮助数据生产单位有效地管理和维护空间数据、建立数据文档，并保证即使其主要工作人员离退时，也不会失去对数据情况的了解；
- 2)提供有关数据生产单位数据存储、数据分类、数据内容、数据质量、数据交换网络及数据销售等方面的信息，便于用户查询检索地理空间数据；

- 3）提供通过网络对数据进行查询、检索的方法或途径,以及与数据交换和传输有关的辅助信息
- 4）帮助用户了解数据，以便就数据是否能满足其需求做出正确的判断;
- 5）提供有关信息，以便用户处理和转换有用的数据。
- 6）根据数据所有者查询所需空间信息

❖元数据对地理信息共享的作用
❖元数据操作工具的作用
❖元数据库的作用

可见，元数据是使数据充分发挥作用的重要条件之一，它可以用于许多方面，包括数据文档建立、数据发布、数据浏览、数据转换等。元数据对于促进数

据的管理、使用和共享均有重要的作用。

- 在地理信息系统中使用元数据，有利于空间数据的管理共享，有利于实现一些特定功能，对于地理信息系统软件的开发，可以提高开发的效率和质量。
- 性能上的原因
- 1）完整性（Completeness）
- 面向对象的地理信息系统和空间数据库的目标之一，是把事物的有关数据都表示为类的形式，而这些类也包括类自身，即复杂的“类的类”结构。这就要求有支持类与类之间相互印证和操作的机制，而元数据可以帮助这个机制的实现。
- 2）可扩展性(Extensibility)
- 有意地延伸一种计算机语言或者数据库特征的语义是很有用的，如把跟踪或引擎信息的生成结果添加到操作请求中，通过动态改变元数据信息可以实现这种功能。
- 3）特殊性(Specialization)
- 继承机制是靠动态连接操作请求和操作体来实现

的，语言及数据库以结构化和语义信息的相关上下文（Context）方式把操作请求传递给操作体，而这些信息可以通过元数据表达。

- 4）安全性(Safety)
- 分类完好的语言和数据库都支持动态类型检测，类的信息表示为元数据，这样在系统运行时，可以被类检测者访问。

- 功能上的原因
- 1）查错功能(Debugging)
- 在查错时使用元数据信息，有助于检测可运行应用系统的解释和修改状态。
- 2）浏览功能(Browsing)
- 为数据的控制类开发浏览器时，为显示数据，要求能解释数据的结构，而这些信息是以元数据来表达的。
- 3）程序生成(Program Generation)
- 如果允许访问元数据，则可以利用关于结构的信息自动生成程序，如数据库查询的优化处理和远程过程调用残体（或“桩”，stub）生成。

- 空间数据元数据标准的建立是空间数据标准化的死前提和保证，只有建立起规范的空间数据元数据才能有效利用空间数据
- 目前，空间数据元数据已经形成了一些区域性或部门性的标准

- FGDC是美联邦地理数据委员会的缩写，它由农业部、商业部、能源部、内务部、众议院、交通部、环保局、国会图书馆、宇航局、档案局等多个部门组成，并由内务部负责，其主要功能是负责联邦地理数据的协调发展、使用、共享和宣传。
- 1994年6月8日批准了Metadata标准。

FGDC于1997年4月又发布了Metadata的修改版，即Metadata2.0版本。

- 国际标准化组织ISO作为全球标准的权威机构，对地理数据标准化问题一直比较重视。成立了地理信息/地理信息业技术委员会，即ISO/TC211，编号为15046，该委员会框架和参考模型(WG1)、地理空间数据模型和算法(WG2)、地理空间数据管理(WG3)、地理空间服务(WG4)、以及行业及功能标准（WG5）5个工作组组成。
- Metadata标准的发展在第三工作组总第15工作小组中进行，即ISO15046-15，地理信息-Metadata工作组。ISO15046-15的目的是想通过建立一个Metadata术语、定义及扩展的公用集合，使地理数据的管理、检索和使用更加有效。

- 欧洲地理信息标准化委员会（CEN/TC287），从1992年，就开始了有关数字地理信息标准化方面的工作，并成立了4个工作组，从地理信息标准化框架（WGI）、地理信息模型和应用（WG2）、

地理信息传输（WG3）以及地理信息定位参考系统（WG4）

- CEN/TC287的Metadata以实用为主，因此它简洁的内容体系受到了国际社会的普遍关注。目前，它正在与ISO/TC211进行有力地合作，以使自己的内容体系能最大程度地被国际社会采纳。

- OpenGIS协会是目前全球最大的地理信息产业化组织，OpenGIS是1993年首先由美国的几个联邦机构和商业组织召开的有关“网络环境下访问异质地理数据和地理处理资源”的会议上提出的概念体系。目前它们已取得了一致，即由ISO/TC211负责制定标准，而OpenGIS协会以该标准为蓝本进行软件实现，

- 元数据标准化逐渐成为共享地学信息的热点
- 元数据标准依赖于信息共享标准的理论

- 空间数据元数据的获取方法
  - 键盘输入：一般工作量大且易出错
  - 关联表：通过公共项（字段）从已存在的元数据或数据中获取有关的
  - 测量法：容易使用且出错较少，如用全球定位系统测量数据空间点的位置等
  - 计算法：由其它元数据或数据计算得到的元数据，如水平位置可由仪器设置及时间计算得到
  - 推理法；根据数据的特征获取元数据。
- 在元数据获取的不同阶段，使用的方法也有差异。

- 1 空间数据元数据库
  - 一般数据集结构简单，分为文本文件

或二维表；元数据可以描述到文件、数据集系列、数据集或数据项

- 空间数据集相对复杂，既有记录空间定位信息的数据文件，又有与该文件连接的属性文件，空间信息之间可能还有拓扑关系；元数据内容繁多，要建立元数据库

❖2 空间数据元数据的组织

❖3 空间数据元数据存储

- 两种存储形式
  - 以数据集为基础，即每个数据集有一个对应的元数据文档，每个元数据文件中包含对应数据集的元数据内容
    - 优点：调用数据时，其相应的元数据也作为一个独立的文件被传输，相对数据库有较强的独立性
    - 缺点：有大量元数据文件，不便于

管理

- 以数据库为基础，一个地理空间数据库有一个元数据文件
  - 优点：便于管理

❖4 空间数据元数据管理

- 地理空间元数据管理模型
- NGDC中地理空间元数据管理模型的设计

❖空间数据元数据的理论和方法涉及到数据库和元数据两方面。

❖由于元数据的内容、形式的差异，元数据的管理与数据涉及的领域有关，它是通过建立在不同数据领域基础上的元数据信息系统实现的。

❖在元数据管理信息系统中，物理

层存放数据与元数据，该层由一些软件通过一定的逻辑关系与逻辑层关联起来。在概念层中用描述语言及模型定义了许多概念，如实体名称、别名等。通过这些概念及其限制特征，经过与逻辑层关联可获取、更新物理层的元数据及数据。

- 元数据系统用于数据库的管理，可以避免数据的重复存储，通过元数据建立的逻辑数据索引可以高效查询检索分布式数据库中任何物理存储的数据。减少数据用户查询数据库及获取数据的时间，从而减低数据库的费用。
- 数据库的建设和管理费用是数据

库整体性能的反映，通过元数据可以实现数据库的设计和系统资源的利用方面开支的合理分配，数据库许多功能（如数据库检索、数据转换、数据分析等）的实现是靠系统资源的开发来实现的，因而这类元数据的开发和利用将大大地增强数据库的功能并降低数据库的建设费用。

- 5 MAPGIS空间数据元数据组织与管理

- 支持多种风格显示

- 元数据附件的添加
- 元数据附件的浏览

- 空间数据共享标准
- 空间数据互操作
- 空间数据交换
- 地理信息Web服务

- 当大量的空间数据出现在网络上，面对多种多样的数据格式，怎样才能有效利用它们，就是空

间数据共享的问题

- GIS互操作是地理信息共享的最高形式，代表着地理信息共享的发展方向

- 空间数据标准化和相互转换，是真正意义上实现空间共享的重要前提。GIS空间数据结构的封闭性，长期以来一直是GIS的瓶颈，每个GIS软件系统必须耗费大量的时间进行数据转换工作，在尚未有统一的空间数据结构标准的前提下，这个问题对WebGIS来说较为突出。解决这个问题的关键是要制定统一的数据结构标准，利用第三方开发的一个高效转换中间界。在国内，尽管不少有识

之士较早就开呼吁建立统一的空间数据标准，由于各方面的原因，这项工作进展缓慢。在国外，这方面较好的标准和工具有:

- （1）空间数据转换标准（SDTS）是由美国地质测量协会（USGS）制定的空间数据在不同计算机系统上的转换标准。它的转换过程是先将数据编码成SDTS文件，再由该SDTS文件转成与SDTS标准兼容的数据格式文件，最后，对文件进行解码便可使用。
- （2)数字地理信息交换标准(DIGEST),它是数字地理信息工作组（DGIWG）的产品。DGIEG由来自北美和欧洲电力总部（SHAPE）组成。DIGEST可以处理栅格、矢量数据的转换(包括拓扑结构)。
- （3）开放的地理数据互操作性规范（OGIS)是由开放的地理信息系统联合

体（Open GIS Consortium）制定的，它主要是提供了一个开发软件的框架，支持数据模型和应用开发，所开发的软件允许用户读取和处理多种地理数据。

- （4）空间档案和交换格式（SAIF）是由 the Government of British Columbia of Canana制定的，它使用一种叫“SAIF/ZIP”的文件格式形成一个通用的转换器，使不同数据格式的数据能方便地转入和转出。
- （5）开放的地理空间数据存储接口（OGDI）是一种新的地理空间数据的标准化读取和转换的技术，实质上是一种应用程序接口（API），它可以将不同数据格式转换成一个统一的数据模型，但是目前只能支持单项转入。

- “互操作地理信息处理”（Interoperable geoprocessing）:
- （1）自由地交换有关地球的信息，即所有关于地表上的、空中的、地球表面以下的对象和现象的信息;
- （2）通过网络协作运行能够操作这些信息的软件。概括为自由交换地理空间信息以及协作运行地理空间信息处理的软件。
- 所谓互操作，就是指异构环境下两个或两个以上的实体，尽管它们实现的语言、执行的环境和基于的模型不同，但它们可以相互通信和协作，以完成某一特定任

务。这些实体包括应用程序、对象、系统运行环境等。

- 数据交换
  - 将一种数据格式转换成为另外某种数据格式的技术
- 空间数据格式转换的内容
  - 空间定位信息
  - 空间关系信息
  - 属性信息

- 空间数据交换方式
  - 外部数据交换模式
  - 直接数据访问模式
  - 基于空间数据转换标准的转换
  - 空间数据互操作模式

万维网地理信息系统

万维网地理信息系统，是GIS与

www的有机结合，GIS通过www功能得到了扩展，从www的任意一个节点，人们可以浏览和获取Web上的各种地理空间数据及属性数据、图像、文件，以及进行地理空间分析，地理数据的概念已扩展为：分布式的、超媒体特性的、相互关联的数据。

- Internet的飞速发展，使传统GIS的发展更加广阔。它改变了GIS数据及应用的访问和传输方式，使GIS真正变成了大众使用的工具。
- WebGIS是Internet和www技术应用于GIS开发的产物，是实现GIS互操作的一条最佳解决途径。从

Internet的任意节点，用户都可以浏览WebGIS站点中的空间数据、制作专题图、进行各种空间信息检索和空间分析。

- WebGIS不但具有大部分乃至全部传统GIS软件具有的功能，而且还具有利用Internet优势的特有功能。这些特有功能包括用户不必在自己的本地计算机上安装GIS软件就可以在Internet上访问远程的GIS数据和应用程序，进行GIS分析，在Internet上提供交互的地图和数据。WebGIS的关键特征面向对象、分布式和互操作。也就是说：任何GIS数据和功能都

是一个对象。这些对象布署在Internet的不同服务器上，当需要时进行装配和集成。Internet上的任何其他系统都能和这些对象进行交换和交互操作。

WebGIS的基本特征

- 1、WebGIS是集成的全球化的客户/服务器网络系统
- 客户/服务器的概念就是把应用分析为服务器和客户两者间的任务，一个客户/服务器应用有3个部分：客户、服务器和网络，每个部分都由特定的软硬件平台支持。客户发送请求给服务器然后服务器处理该请求，并把结果返回给客户，客户再把结果或数据提供给用户。

客户和服务器间的连接根据TCP/IP这样的能信协议来建立。

- WebGIS应用客户/服务器概念来执行GIS的分析任务，它把任务分为服务器端和客户端两部分，客户可以从服务器请求数据、分析工具和模块，服务器或者执行客户的请求并把结果通过网络送回给客户，或者把数据和分析工具发送给客户供客户端使用。

- 2、WebGIS是交互系统
- 通过超链接（Hyperlink），www提供在Internet上最自然的交互性，用户通过超链接，可以一页一页地浏览Web页面。然而，每个Web页面是由WWW开发者组织

的静态图形和文本所组成。这些图形大部分是JPG和GIF格式的文件，因此用户无法操作地图，甚至连像Zoom、Pan、Query这样简单的分析功能都无法执行。WebGIS却可使用户在Internet上操作GIS地图和数据，用Web浏览器执行像Zoom、Pan、Query和Label这样的基本GIS功能，甚至可以执行像“离你最近的旅馆或饭店在哪儿”这样的空间查询，或者更先进的空间分析，比如缓冲分析和网络分析等，在Web上作用WebGIS就和在本地计算机上使用桌面GIS软件一样。

- 3、WebGIS是分布式系统
- Internet的一个特点就是它可以访问分布式数据库和执行分布式处理，即信息和应用可以部署在跨越整个

Internet的不同计算机上。WebGIS利用Internet这种分布式系统把GIS数据和分析工具部署在网络不同的计算机上。GIS数据和分析工具是独立的组件和模块，用户可以随意从网络的任何地方访问这些数据和应用程序。用户不需要在自己的本地计算机上安装GIS数据和应用程序，只要把请求发送到服务器，服务器就会把数据和分析工具模块传送组用户，达到Just-in-Time的性能。

- 4、WebGIS是动态系统
- 由于WebGIS是分布式系统，数据库和应用程序部署在网络的不同计算机上，并由其管理员进行管理，因此，这些数据和应用程序一旦由其管理员进行更新，则它们对于Internet上的每个用户来说都将是最新可用的数据和应用。这也

就是说，WebGIS和数据源是动态链接的，只要数据源发生变化，WebGIS将得到更新。和数据源的动态链接将保持数据和软件的现势性。

- 5、WebGIS是跨平台系统
- WebGIS可以访问不同的平台，而不必关心用户运行的操作系统是什么（如Windows、UNIX、Macintosh）。WebGIS对任何计算机和操作系统都没有限制。只要能访问Internet，用户就可以访问和使用WebGIS。随着Java，.Net语言技术的发展，未来的WebGIS可以做到“一次编写，到处运行”，使WebGIS的跨平台特性走向更高

层次。

- 6、WebGIS能访问Internet异构环境
- 在GIS用户组间访问和共享GIS数据、功能和应用程序，需要很高的互操作性。开放式地理数据互操作规范（Open Geodata Interoperablity Specification）为GIS互操作性提出了基本的规则。其中有很多问题需要解决，例如数据格式的标准、数据交换和访问的标准，GIS分析组件的标准规范等。随着Ineternet技术和标准化的飞速发展，完全互操作的WebGIS将会成为现实。

- 7、WebGIS是图形化的超媒体信息系统
- 使用Web上超媒体系统技术，WebGIS通过超媒体热链接可以链接不同的地图

页面。例如，用户可以在浏览全国地图时，通过单击地图上的热链接，而进入相应的省地图进行浏览。

- 另外，WWW为WebGIS提供了集成多媒体信息的能力，把视频、音频、地图、文本等集中到相同的Web页面，极大地丰富了GIS的内容和表现能力。

空间数据库设计

主讲：赵娜
zhaona@tjau.edu.cn

- 空间数据库设计概述
- 需求分析
- 概念结构设计
- 逻辑结构设计
- 空间数据库物理设计
- 空间数据库的实施和维护
- 空间数据库建库
- 武汉市江夏区土地利用规划空间数据库库设计

- 根据具体地理空间数据库应用目的和工程要求，在一个特定的应用环境中，确定能被一定的空间数据库管理系统接受的最优数据模型、处理模式、存储结构和存取方法，实现对应用系统有效的管理，满足用户信息要求和处理要求。

- 数据模型
  - 实体及其相互关系的数学描述，是空间数据库建立的逻辑模型
    - 层次模型、网络模型、关系模型
- 处理模式
  - 空间数据的处理方法
- 存储结构
  - 空间数据的物理组织
- 存取方法

- 空间数据的存取

✓ 空间数据库设计与应用系统设计相结合原则
✓ 数据独立性原则
✓ 共享度高，冗余度低
✓ 用户与系统的接口简单性原则
✓ 系统可靠性、安全性与完整性原则
✓ 系统具有重新组织、可修改与可扩充原则

❖ 空间数据库系统的生存期
- 从开始规划、设计、实现、维护到最

后被新的系统取代而停止使用的整个期间。

- 生存期划分为6个阶段
  - 需求分析
  - 概念设计
  - 逻辑设计
  - 物理设计
  - 实现
  - 运行和维护

数据库设计步骤

- 需求分析阶段，综合各个用户的应用需求；
- 在概念设计阶段形成独立于机器

特点，独立于各个DBMS产品的概念模式(E-R图)；

- 在逻辑设计阶段将E-R图转换成具体的数据库产品支持的数据模型，如关系模型，形成数据库*逻辑模式*；然后根据用户处理的要求、安全性的考虑，在基本表的基础上再建立必要的视图(View)，形成数据的*外模式*；
- 在物理设计阶段，根据DBMS特点和处理的需要，进行物理存储安排，建立索引，形成数据库*内模式*。

- **需求分析的任务**
  - 调查的重点是“**数据**”和“**处理**”，

通过调查、收集与分析，获得用户对数据库的如下要求：

- (1) **信息要求**。指用户需要从数据库中获得*信息的内容*与*性质*。由信息要求可以导出数据要求，即在数据库中需要存储哪些数据。
- (2) **处理要求**。指用户要完成什么*处理功能*，对处理的*响应时间*有什么要求，*处理方式*是批处理还是联机处理。
- (3) **安全性**与**完整性**要求。 确定用户的最终需求是一件很困难的事设计人员必须不断深入地与用户交流，才能逐步确定用户的实际需求。

❖常用的调查方法有：

- (1) 跟班作业。通过亲身参加业务工作来了解业务活动的情况。这种方法可以比较准确地理解用户的需求，但

比较耗费时间。

- (2) 开调查会。通过与用户座谈来了解业务活动情况及用户需求。座谈时，参加者之间可以相互启发。
- (3) 请专人介绍。
- (4) 询问。对某些调查中的问题，可以找专人询问。
- (5) 设计*调查表请用户*填写。如果调查表设计得合理有效，也易于为用户接受。
- (6) 查阅记录。查阅与原系统有关的数据记录。

❖ 空间数据需求分析主要包括的内容

- 用户基本需求调研
- 空间数据现状分析
- 系统环境/功能分析

- 概念结构设计
  - 对用户信息需求的综合分析、归纳，形成一个不依赖于空间数据库管理系统的信息结构设计
- 设计原则
  - 语义表达能力强
  - 易于用户理解
  - 独立于具体的物理实现
  - 易于向数据模型转换
- 建模工具
  - E-R模型
  - UML模型
  - 面向实体模型

- ER模型是最为流行的建模工具之一。
- 实体和属性
  - 实体是物理上或者概念上独立存在的事物或对象。

- 实体由属性来刻画性质，属性可以是单值或多值的。

➢ 联系

- 一对一（1：1）
  - 在一对一的联系中，一个实体中每个实例只能与其他参与实体的一个实例相联系。
  - 例如，实体MANAGER和FOREST之间的联系manages就是一个一对一的联系。

- 多对一（M：1）
  - 多对一联系可将一个实体的多个实例与另一个参与该联系的实体的一个实例相连接。
  - Belongs_to是实体FACILITY与FOREST之间的一个多对一联系，这里假定每个设施仅仅属于一个森林，但每个森林可以有多个设施。

- 多对多（M：N）

- 一个实体的多个实例会与另一个参与该联系的实体的多个实例相联系。
- 实体RIVER和FACILITY之间的联系supplies_water_to正是这样的一个联系。

❖ 实体——矩形
❖ 属性——椭圆
  - 用直线与实体矩形连接
  - 多值属性用双椭圆
  - 码的属性加下划线
❖ 联系——菱形
  - 联系的基数标注在菱形的旁边

❖ 实体
  - forest、forest-stand、river、

road、facility

- fire-station、manager

❖ 联系

- 固有的联系
  - cross、within、part-of
- 隐含的联系
  - 例如：一条河流穿过道路是标明的，但是一条河流穿过森林是隐含的

➢ 前面的ER图只是将空间属性转化为空间表，只是把空间属性简单的当做非空间属性处理

➢ 没有充分利用空间数据模型

❖ 实体象形图

- 象形图
  - 象形图是一种将对象插在方框内的微缩图表示，这些微缩图用来扩展

ER图，并插到实体矩形框中的适当位置。

- 可以是基本的形状，也可以是用户自定义形状

- 形状
  - 形状是象形图中的基本图形元素，它代表着空间数据模型中的元素。
  - 一个模型元素可以是基本形状、复合形状、导出形状或备选形状。

❖ 基本形状

❖ 复合形状

- 为了处理那些不能用某个基本形状表示的对象，定义了一组聚合的形状，并用基数来量化这些复合形状。
- 例如,河流网可以用线的象形图的连

接表示，且其基数为n

- 对于无法在某个给定比例尺下描绘的要素，用基数0表示

❖ 导出形状

- 如果一个对象的形状是由其他对象的形状导出的，那么就用斜体形式来表示这个象形图。
- 例：从美国的州界导出美国的形状

❖ 备选形状

- 备选形状可以用于表示某种条件下的同一个对象。

- 例如，根据比例尺，一条河流可以表示成一个多边形或一条线。

<备选形状>　　<基本形状>
<导出形状>

<　基本形状>　<基本形状>

备选形状的语法

●　〗　　●　〗

备选形状的象形图

❖任意形状

- 对于形状的组合，用通配符（*）表示，它表示各种形状。
- 例如，一个灌溉网是由泵站（点）、水渠（线）以及水库（多边形）所组成的。

❖用户自定义形状

- 除了点、线和多边形这些基本形状外，用户还可以定义自己的形状。
- 例如，为了表达更多的信息，用户可能更愿意使用感叹号之类的象形图来表示灌溉网。

❖ 联系象形图

- 联系象形图用来构建实体间联系的模型。
- 例如，part-of用于构建道路与路网之间联系的模型，或是用于把森林划分成林分的建模。

Part_of（网络）　　Part_of（分区）联系的象形图

❖UML内容

- UML语义:描述基于UML的精确元模型定义
- UML表示方法 :五类图形(共9种)
  - 用例图
  - 静态图(Static diagram)，包括类图、对象图和包图
  - 行为图(Behavior diagram) ，包括状态图和活动图
  - 交互图(Interactive diagram) ,包括顺序图和合作图
  - 实现图(Implementation diagram)，包括构件图和配置图

❖ UML本身是一个完整的建模语言，支持系统开发的不同阶段，从需求分析到系统测试;

- 需求分析阶段：UML通过用例图（有时也需一些简单的类图、活动图）来捕获用户需求，描述对系统感兴趣的外部角色和他们对系统的功能要求;
- 系统分析阶段:主要关心问题域的概念和实

体，并得到与问题域直接相关的类和对象，以及它们之间的关系（类图、顺序图、协作图、状态图、活动图）；

- 设计阶段：需要定义一些与技术实现相关的类，如：用户接口、数据库、通信和并行等问题，UML提供了强大的静态和动态建模机制（类图、顺序图、协作图、状态图、活动图、组件图、实施图）
- 实现阶段：类---->语言代码；
- 单元测试阶段：依据类图和类的规格说明
- 集成测试阶段：测试人员依据构件图和合作图；
- 系统测试阶段：测试人员主要依据用例图来验证系统的行为；

- **逻辑结构设计的任务**就是把概念结构设计阶段设计好的*基本E-R图*转换为与选用DBMS产品所支持的数据模型相符合的*逻辑结构*。
- 实际情况往往是已给定了某种DBMS，设计人员没有选择的余地。
- 设计逻辑结构时一般要分三步进行：
  - (1)将概念结构转换为一般的关系、网状、层次模型；
  - (2)将转换来的关系、网状、层次模型向特定DBMS支持下的数据模型转换；
  - (3)对数据模型进行优化。

- 数据模型
  - 传统数据模型
    - 层次模型
    - 关系模型
    - 网络模型
  - 面向对象数据模型
  - 空间数据模型
    - 混合数据模型
    - 集成数据模型
    - 地理关系数据模型

- 利用一组文件形式来存储地理数据中的空间数据及其拓扑关系数据，利用通用关系数据库的标准来存储属性数据，通过一个唯一的标识符来建立它们之间的关

联。

- 例如
  - Arc/Info
  - MapInfo

❖缺点

- 空间数据与属性数据分开存储,缺乏完整的存储机制

❖纯关系数据模型

❖空间数据和属性数据都用关系数据库的标准来存储，是用标准关系联结机制建立空间数据与属性数据的关联

- SDE

✓优点

- 保持了数据的完整性

×缺点

- 数据类型存在局限性,用户不能自定义数据字段，缺乏空间查询语言

- 利用面向对象的技术来建立GIS软件，从而提出了面向对象数据模型
- 特点
  - 建立子对象层次结构,符合地理要素层次分类
  - 便于应用开发和数据建模
- 目前面向对象GIS的两种趋势
  - 建造纯面向对象数据存储
  - 在关系数据库的基础之上建立存储对象的机制

- 空间逻辑模型分类
  - 结构化模型
    - 层次模型

        - 网络模型
    - 面向操作模型
        - 关系模型

- 在构建模型的过程中，空间实体及其相互关系的表达是研究重点。要对空间实体进行更加详细的建模，并对其元素对象的属性和方法进行定义

- 所有几何实体都是由点和弧段组成的
- 空间数据
    - 点
    - 弧段
- 几何实体是由空间数据和图形信息组成的

- 要素的表示结构

- 几何形态
  - 点、线、区以及复合
- 属性

❖ 几何表示结构

- 点几何实体
- 线几何实体
- 区几何实体

❖ 空间数据的表示结构

- 点
- 弧段

❖ 标注

- 在制图显示时用来标注要素的文本，可以确定位置或识别要素

❖ 标注类型

- 静态注记
- 属性注记

- 维注记

❖ 空间数据库物理设计
  - 物理表示组织
  - 空间数据的存取

❖ 地理数据库数据字典表关系
❖ 要素类的关系模式
❖ 注记类的关系模式
❖ 关系类的关系模式
❖ 规则的关系模式

❖ 实施

- 数据录入
- 数据编辑
- 数据库建立
- 数据分析与处理
- 数据输出

❖ 维护
- 程序的维护
- 数据文件的维护
- 代码的维护
- 机器、设备的维护

❖ 维护类型
- 更正性维护
- 适应性维护
- 完善性维护
- 预防性维护

❖ 资料准备和预处理

❖ 数据采集

- 数据处理
- 空间数据库建库

- 资料是获取数字地形图数据的基础
- 数字化作业前应该收集的资料内容
  - 最新出版的地形图、航片、卫片和影像资料
  - 最新出版的中华人民共和国行政区划代码、世界各国和地区名称代码、行政区划简册、数字化区域的地名录
  - 最新出版的交通资料和国家公布的交通信息
  - 实测或编绘出版的最新基本比例尺地形图及有关现势资料
  - 国家、省、县、乡公路线路名称和编码
  - 中华人民共和国铁路路线名称代码
  - 中华人民共和国铁路车站站名代码
  - 全国河流名称代码
  - 成图区域内的测量控制点成果

- 资料内容

- 资料精度
- 资料现势性
- 资料介质
- 资料形式

- 地图资料要查明地图的出版机关、出版年代、比例尺、成图方法、精度、采用资料的来源、数学基础、各要素内容与现状的符合程度、采用的图式及特点说明
- 航片、卫片和影像资料要查明摄影参数
- 对参考资料分析着重研究资料来源的可信度、内容的现势性和完整性
- 对补充资料分析着重研究出版机关、年代和特点及转标这些内容

的方法，如交通图

- 掌握成图区域的地理景观和地理特征

- 图面预处理
- 属性数据处理

- 数据采集方法
  - 扫描数字化地图
  - 遥感影像提取数据
  - 卫星定位系统和测量仪器外业数据采集
  - 利用空间数据编辑处理功能以人机交互方式采集空间数据同时录入必要的属性数据

- 常用的数字化方法

- 手扶跟踪数字化
- 扫描数字化
- 屏幕数字化

❖地图数字化流程

❖错误检查与编辑
❖几何纠正
❖地图投影转换
❖图形解译
❖投影转换
❖图幅拼接
❖拓扑关系生成
❖属性数据录入

❖步骤
- 数据字典和数据索引的生成
- 图形与属性数据库的建立

- 设立用户密码、规定用户使用权限
- 软件系统与数据的融合检查
- 数据库系统试运行测试