



# 地理空间数据库

陶煜波

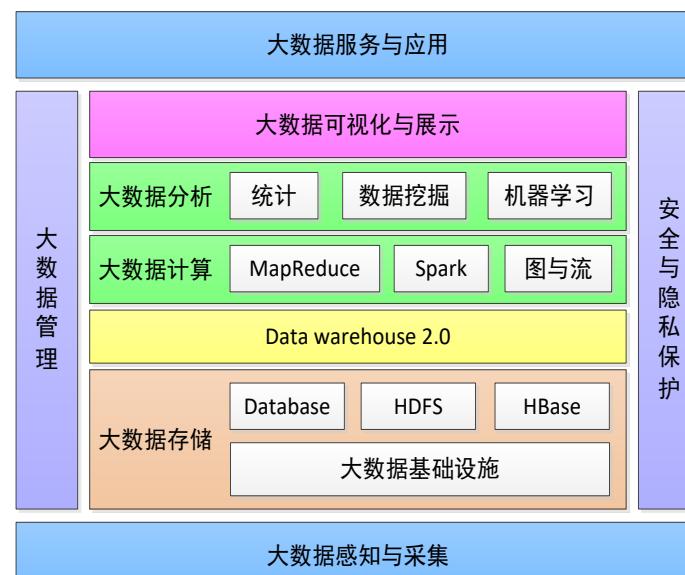
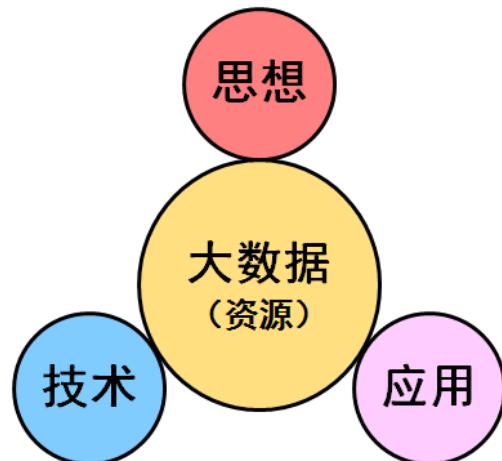
计算机科学与技术学院

[taoyubo@cad.zju.edu.cn](mailto:taoyubo@cad.zju.edu.cn)



# 大数据

- “当今世界，科技进步日新月异，**互联网、云计算、大数据**等现代信息技术深刻改变着人类的思维、生产、生活、学习方式，深刻展示了世界发展的前景。”
- 大数据**既是一类**数据**，也是一项**技术**，还是一种**理念**



# 大数据

- Data contains **value** and **knowledge**
- But to extract the knowledge from data needs to be
  - Stored
  - Managed
  - Analyzed
  - Visualized
- Michael Stonebraker
  - 大数据量、结构化、“小分析学”
  - 大数据量、非结构化、“大分析学”
  - 大速度
  - 大多样性



Won Turing award 2015



# 大数据

- 大数据量、结构化、“小分析学”
  - 没有人会用**select \***来查询，因为无法处理返回的**TB**级数据。替代方案是**SQL**的聚集分析，如结合**group by**的**count**、**sum**、**max**、**min**、**avg**等
- 大数据量、非结构化、“大分析学”
  - 使用回归、分类、聚类、搜索、推荐等数据挖掘和机器学习方法
- 大速度
  - 对电子交易、实时网页广告投放、实时客户针对营销、移动社交网络等应用，处理“灭火水龙带”式涌入的数据
- 大多样性
  - 电子表格、网页、**XML**、关系型数据库、几何等数据格式

# 空间数据

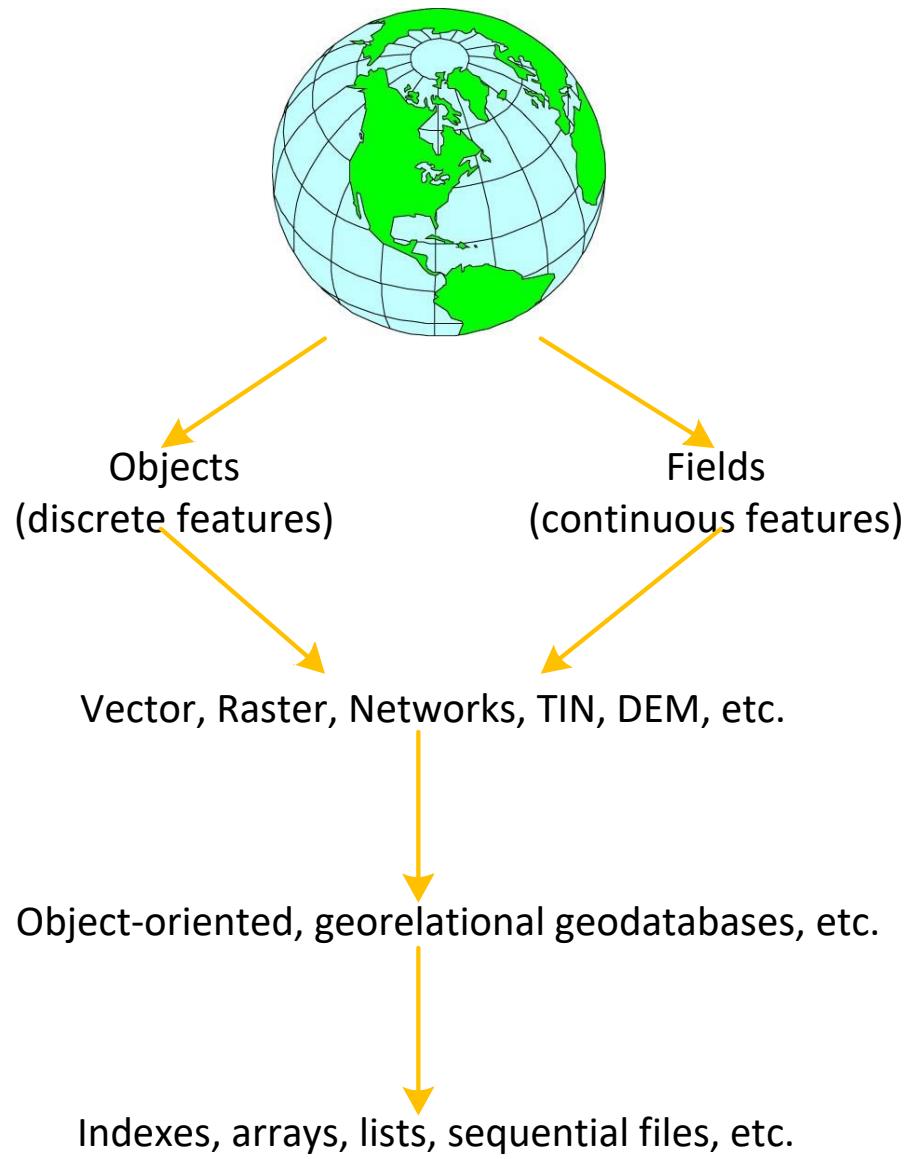
## 1. Reality

## 2. Information Model (conceptual model)

## 3. Representation or Data Model (logical model)

## 4. Databases (physical model)

## 5. File Structures (binary model)



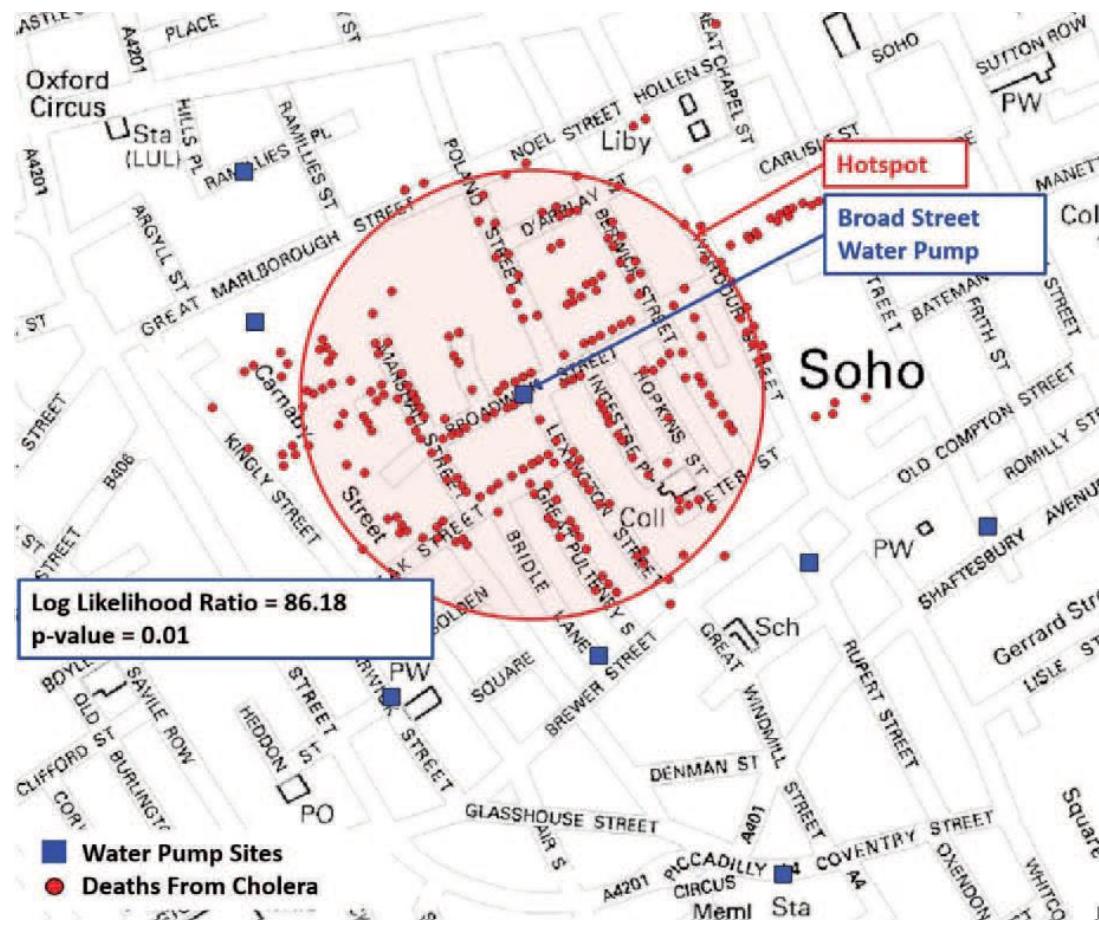
# 空间分析

| Spatial Relationship | Description   |
|----------------------|---|
| Measurements<br>度量   | What are the physical distances, lengths, and area?<br>- patterns, distance, perimeter, area  |
| Proximal<br>邻近       | Are within a distance of?<br>- near, far, absolute, relative  |
| Topological<br>拓扑    | Are spatially identical to? Identical means same shape, size on top of each other, same vertices, and same location<br>- partially equivalent (intersect), within (contained/ adjacent) |
|                      | Are spatially adjacent to?<br>- neighbours  |
| Directional<br>方向    | Bearings and compass directions<br>- north, south, east, west   |



# 空间分析举例

- 1854年，约翰·斯诺(John Snow)医生手工绘制了标有霍乱病例位置的伦敦街道图，以可视的形式标识出了Broad Street水泵附近的疾病爆发热点区域

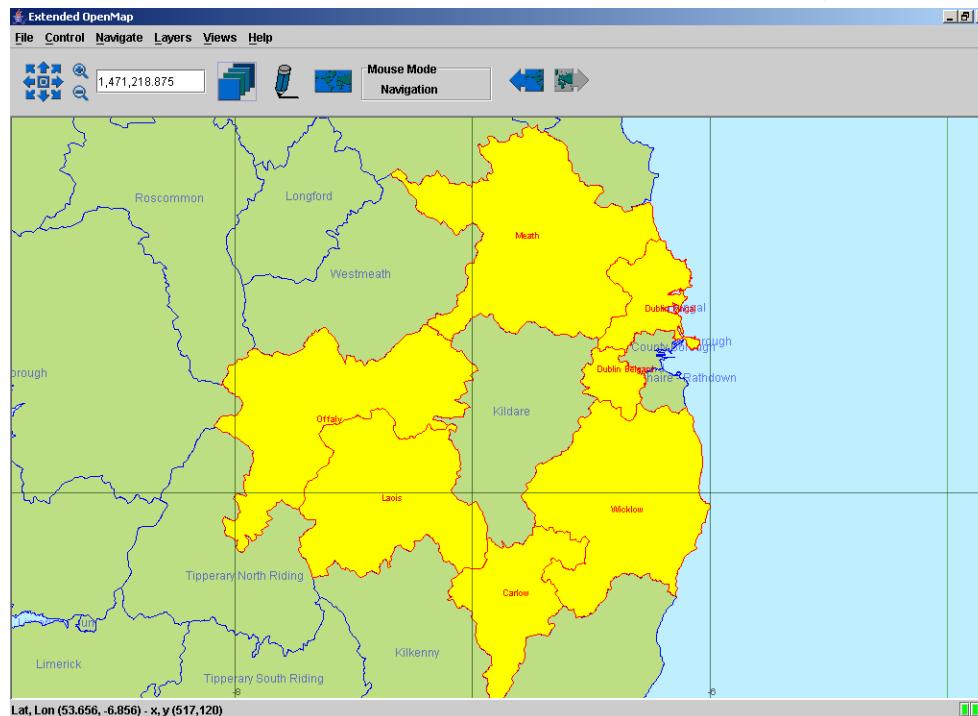


# 空间查询举例

关系数据库(PostgreSQL): select, from, where, and  
空间操作(PostGIS): ST\_Touches  
用户自定义关系: County(name, the\_geom, ...)

- 所有和Kidare接壤的郡 (ArcMap)

```
SELECT C1.Name  
FROM County C1, County C2  
WHERE ST_Touches(C1.the_geom, C2.the_geom)  
      AND C2.Name = 'Kidare';
```





# 空间查询举例

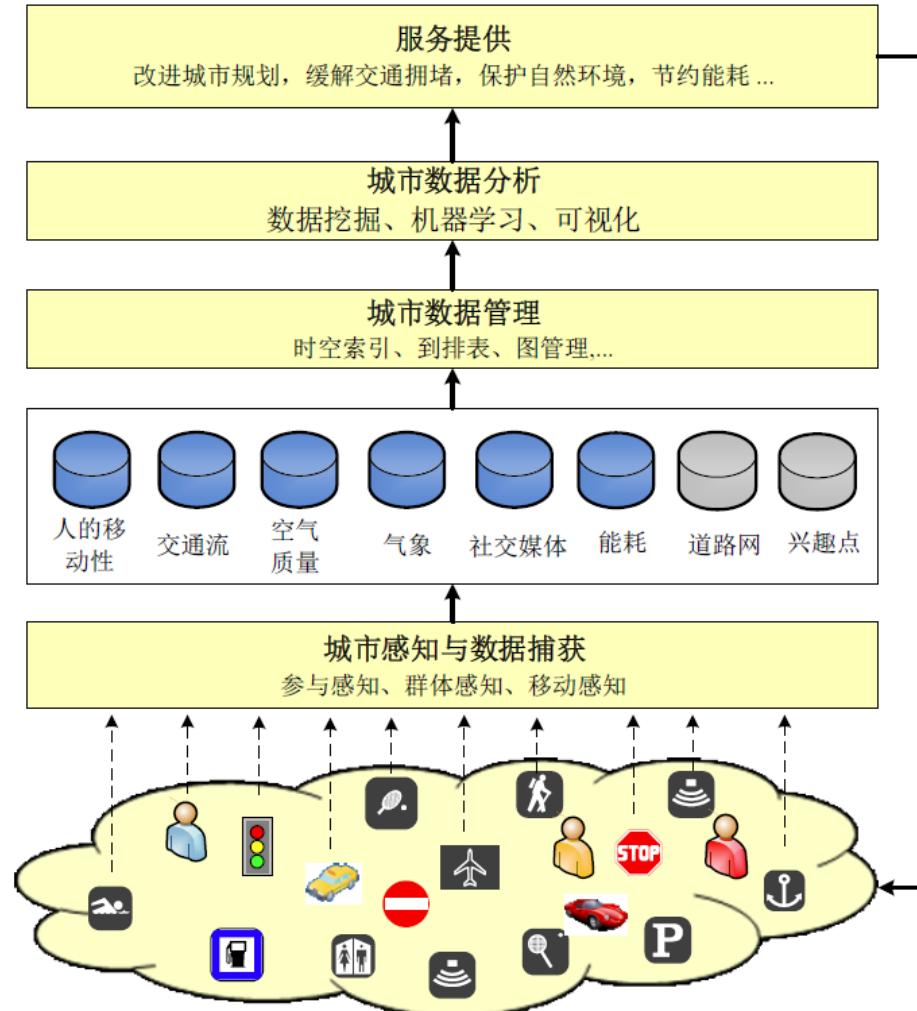
- 健康码：时空查询，接触网络查询



那么我们常用的健康码是怎样工作的呢

# 空间大数据 – 城市计算

- 核心问题
  - 数据感知与捕获
  - 海量异构数据的管理
  - 异构数据的协同计算
  - 虚实结合的混合式系统
- 应用
  - 城市规划、智能交通
  - 环境、能源消耗
  - 社交和娱乐、经济
  - 城市安全和应急响应
- 课程核心: **空间计算**
  - 空间关联、时空查询



以城市计算为例



# 空间计算 (Spatial Computing)

- Spatial Computing
  - A set of ideas and technologies that transforming our lives by **understanding** the physical world, **knowing** and **communicating** our **relation** to **places** in that work, and **navigating** through these places
  - Applications
    - A Didi driver knows precisely where customers are, nearby points of interest, and how to reach their destinations
    - Groups of friends can form impromptu events via "check-in" models used by Facebook and foursquare
    - Scientists use GPS to track endangered species to better understand behavior
    - Farmers use GPS for precision agriculture to increase crop yields while reducing costs
    - Google Earth is being used to teach children about their neighborhoods and the world
    - Augmented reality applications (e.g., Pokemon Go) are providing real-time place labeling in the physical world and providing people detailed information about major landmarks nearby



# 空间计算举例

## Planning Bike Lanes based on Sharing-bikes' Trajectories

Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, Yu Zheng



# 空间计算举例

- 时空数据：信息安全与国家安全

图灵的猫 bilibili

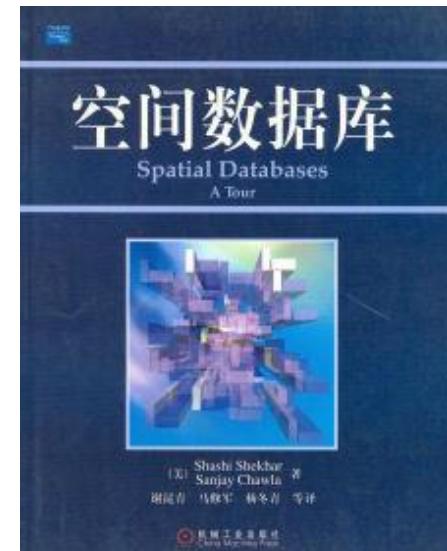
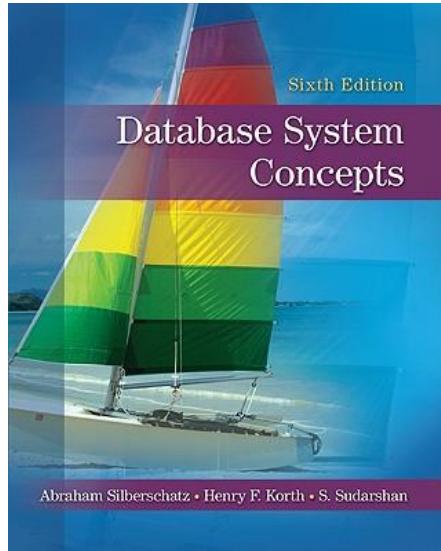
# 课程安排 – 理论 应用 实现

| 日期    | 课程内容                   | 作业            | 日期    | 课程内容                   | 作业      |
|-------|------------------------|---------------|-------|------------------------|---------|
| 9.14  | 地理空间数据库概论<br>关系模型与关系代数 |               | 11.16 | 空间查询处理与优化              | Quiz 4  |
| 9.28  | 关系数据库标准语言              | SQL<br>Quiz 1 | 11.23 | 空间网络模型<br>pgRouting    | Network |
| 10.9  | 几何对象模型                 | SSQL          | 11.30 | 视图与触发器                 | Quiz 5  |
| 10.12 | PostGIS空间几何查询          | Quiz 2        | 12.7  | PostgreSQL函数与触发器       | App     |
| 10.19 | 数据库设计<br>空间扩展E-R图      | Design        | 12.14 | 联机事务处理                 | Quiz 6  |
| 10.26 | 关系数据库设计理论              | Quiz 3        | 12.21 | OLAP, NoSQL等<br>Quiz 7 |         |
| 11.2  | 空间存储与索引                |               | 12.28 | 期末复习                   |         |
| 11.9  | 空间索引                   | Index         |       |                        |         |

- 数据库系统原理
- 地理空间数据库

# 参考教材

- Database System Concepts (第六版)
  - Abraham Silberschatz, Henry F.Korth, and S.Sudarshan
- 空间数据库管理系统概论
  - 程昌秀 编著
- Spatial Databases: A Tour
  - Shashi Shekhar and Sanjay Chawla



# 网络课程

---

- Stanford Online Course
  - Introduction to Databases
  - <https://online.stanford.edu/courses/soe-ydatabases-databases>
  - Quiz, Interactive Exercise, Exam
  
- Minnesota Course
  - Spatial Data Science
  - <http://www.spatial.cs.umn.edu/Courses/Fall21/5715/>
  - Chapter 1-6, Quiz, Homeworks, Labs

# 课程考察

---

- 作业占最终考评成绩的40%
  - 6个数据库查询、设计、优化、实现作业
  - 截止日期后n ( $n < 10$ )天内提交的作业分数  
最终作业分数 = 作业得分  $\times (1 - n/10)$
  - 附加题作为额外加分
- 测试占最终考评成绩的10%
  - 6个学在浙大在线测试，1个课堂书面测试
- 期末闭卷考试占最终考评成绩的50%
  - 可带一张A4纸
- 平时成绩 = 作业分数 + 测试分数
- 最终成绩 = (平时成绩 + 期末成绩) / 2

# 课程学习建议

---

- 积极提问和回答问题，批判性思考
- 查询PostgreSQL、PostGIS、pgRouting等帮助文档
- 作业很“难”，也很重要
  - 尽早开始，每次作业可能需要12-24个小时
  - 数据驱动的问题，生活实际案例，尽可能考虑全面
  - 培养自己分析问题、用SQL语句解决问题的能力
  - 可以相互讨论思路，作业抄袭或雷同都要扣分
  - Have Fun
- 认真准备测试和期末考试
- 希望学些有用的知识，做些有用的事情

# 第一章 地理空间数据库概论

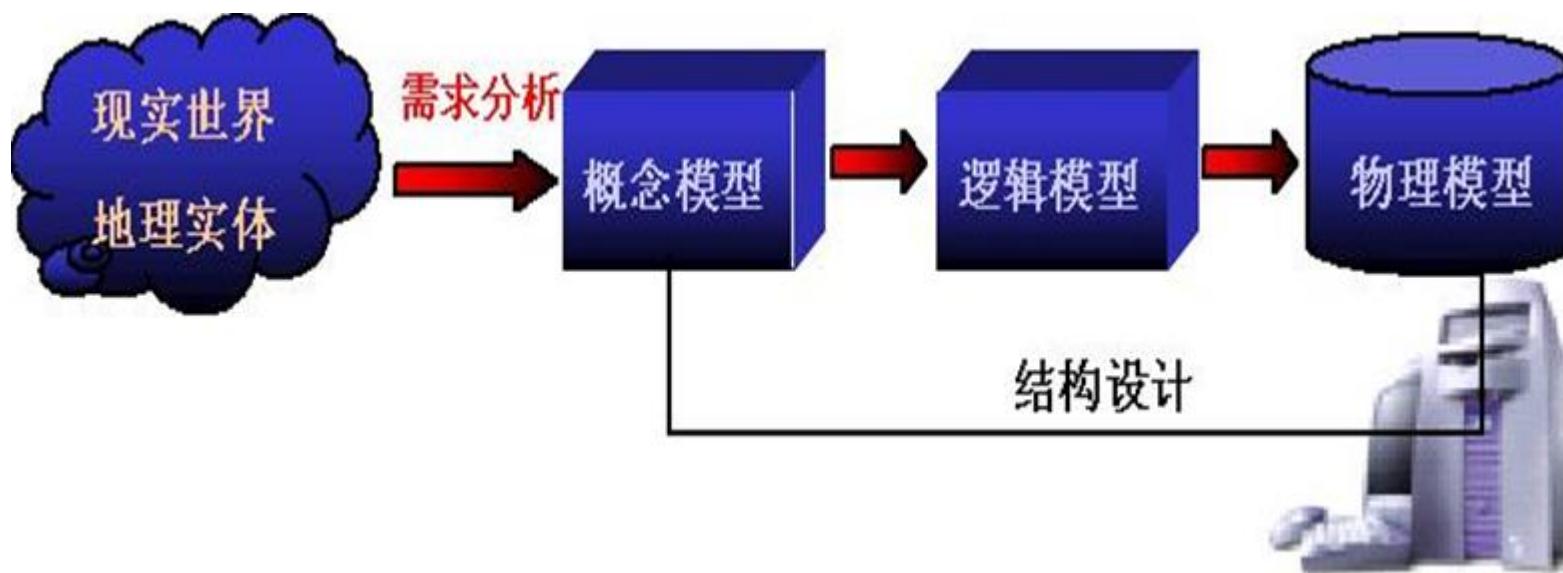
---

- 1.1 关系数据库基本概念
  - 1.1.1 数据模型
  - 1.1.2 三级模式
- 1.2 空间数据库基本概念
- 1.3 空间数据管理技术的产生与发展
- 1.4 现有空间数据库标准简介
- 1.5 现有空间数据库管理系统产品简介 (自学)

参考教材：  
数据库系统概念 1.1-1.14  
空间数据库管理系统概论 1.1-1.6

## 1.1.1 数据模型

- 数据模型是用来**描述**数据、**组织**数据和对数据进行**操作**的模型



# 概念数据模型

---

- 概念数据模型(**conceptual model**)独立于计算机系统，用来描述某个特定组织所关心的信息结构，按用户的观点来对数据和信息建模，对企业主要数据对象的基本表示和概括性描述，主要用于**数据库设计**
- 强调其语义表达能力，概念应该简单、清晰，易于用户理解，是数据库设计人员和用户之间进行交流的工具
- 概念数据模型与 DBMS (**Database Management System**)无关

# 逻辑数据模型

---

- 逻辑数据模型(logical model)直接面向数据库的逻辑结构，通常有一组严格定义的，无二义性的语法和语义的数据库语言，人们可以用这种语言来定义、操纵数据库中的数据
- 逻辑数据模型与DBMS有关，DBMS以所支持的逻辑数据模型来分类
- 用概念数据模型表示的数据必须转化为逻辑数据模型表示的数据，才能在DBMS中实现
- 逻辑数据模型既要面向用户，也要面向实现

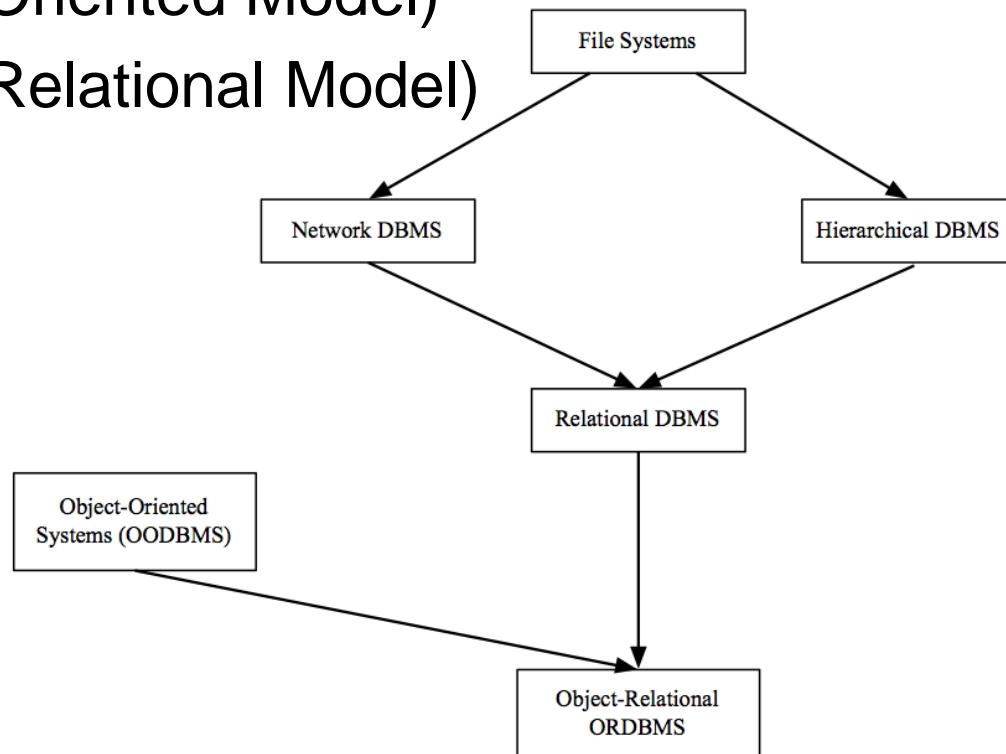
# 物理数据模型

---

- 物理数据模型(**physical model**)是对数据最低层的抽象，它描述数据在磁盘或磁带上的存储方式和存取方法，是面向计算机系统的
- 每种逻辑数据模型在实现时，都有其对应的物理数据模型
- 物理数据模型的实现不但与**DBMS**有关，还与操作系统和硬件有关

# 数据模型

- 逻辑数据模型
  - 层次模型 (Hierarchical Model)
  - 网状模型 (Network Model)
  - 关系模型 (Relational Model)
  - 面向对象模型 (Object Oriented Model)
  - 对象关系模型 (Object Relational Model)



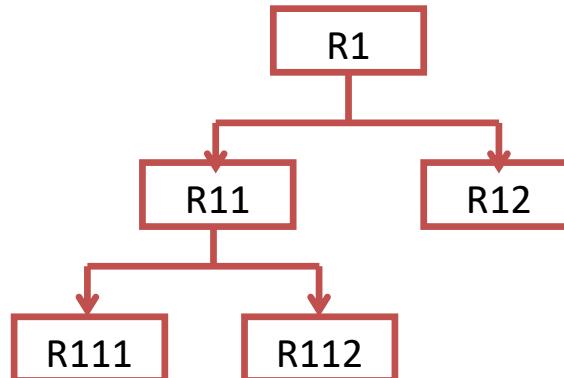
# 数据模型

---

- 数据模型
  - 数据结构     $\leftrightarrow$  静态特性
    - Rows and columns?
    - Nodes and edges?
    - Key-value pairs?
    - A sequence of bytes?
  - 数据操作     $\leftrightarrow$  动态特性
    - Find the value of key x
    - Find the rows where column ‘lastname’ is ‘Jordan’
    - Get the next N bytes
  - 完整性约束  $\leftrightarrow$  完整性约束条件
    - All rows must have the same number of columns
    - All values in one column must have the same type
    - A child cannot have two parents

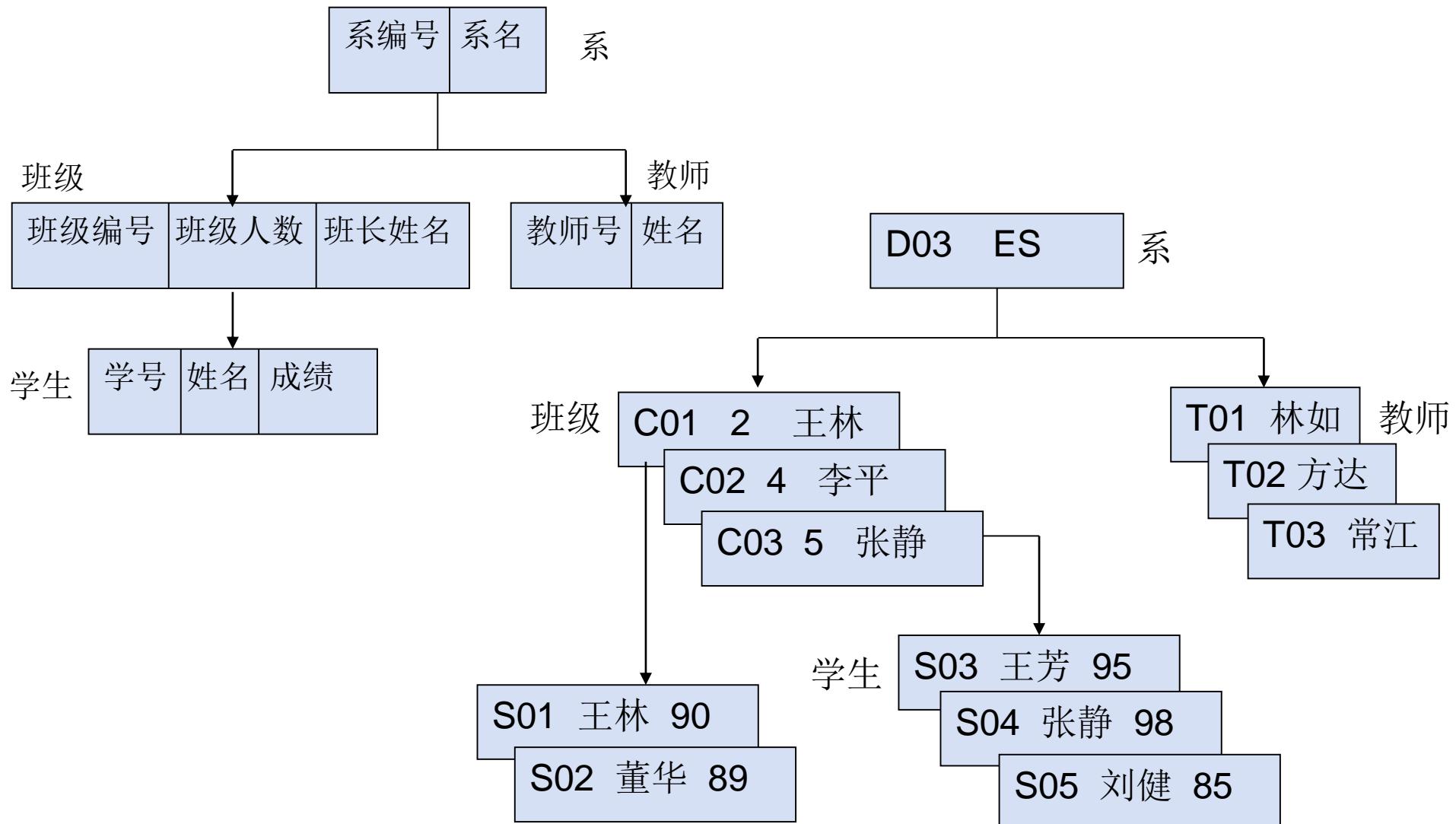
# 层次模型

- 数据结构 – 树
  - 有且只有一个结点没有双亲结点，这个结点称为根结点
  - 根以外的其它结点有且只有一个双亲结点
- 数据操作
  - 查询、插入、删除、更新
- 完整性约束
  - 无相应的双亲结点值就不能插入子女结点值
  - 如果删除双亲结点值，则相应的子女结点值也被同时删除



# 层次模型

- 学生-教师层次数据模型和实例



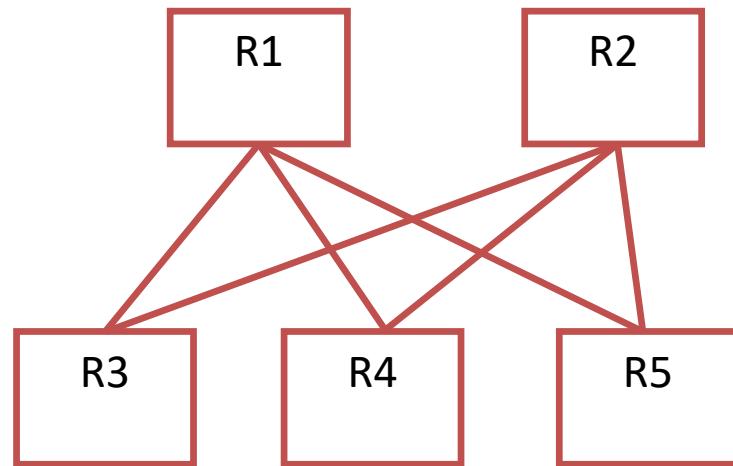
# 层次模型

---

- 优点
  - 层次数据模型简单，对具有一对多的层次关系的部门描述自然、直观，容易理解
  - 性能优于关系模型，不低于网状模型
  - 层次数据模型提供了良好的完整性支持
- 缺点
  - 多对多联系表示不自然
    - 学生与课程之间的选课关系
  - 对插入和删除操作的限制多
  - 查询子女结点必须通过双亲结点
    - 查询有多少学生成绩在90分以上
  - 层次命令趋于程序化

# 网状模型

- 数据结构 – 图
  - 允许多个结点没有双亲结点
  - 允许结点有多个双亲结点
  - 允许两个结点之间有多种联系 (复合联系)
- 数据操作
  - 查询、插入、删除、更新
- 完整性约束



# 网状模型

---

- 优点
  - 能够更为直接地描述现实世界，如一个结点可以有多个双亲
  - 具有良好的性能，存取效率较高
- 缺点
  - 结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于最终用户掌握
  - DDL (Data Definition Language)、DML (Data Manipulation Language)语言复杂，用户不容易使用

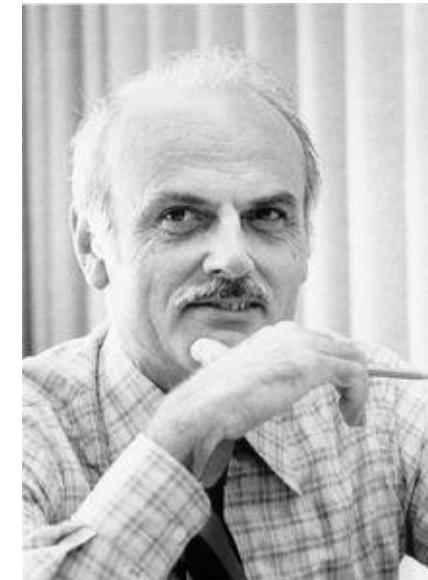
# 关系模型

---

- 和层次数据模型和网状数据模型相比，关系数据模型的主要优点
  - **简单**，一个数据库由多个关系组成，每一个关系就是一个规范化了的二维表，关系模型中的许多概念与二维表是一一对应的，普通用户容易理解
  - **易访问**，可以使用高级的数据查询语言构造出复杂的查询，对数据库中的数据进行访问
- 40多年来，关系模型无论在理论研究还是在系统研制方面均取得了辉煌的成就
- 关系模型从实验室走向了社会，涌现出许多性能良好的商品化关系数据库管理系统(简称RDBMS)
  - 例如Oracle, MySQL, MS SQL Server, PostgreSQL, DB2等

# 关系模型

- 关系数据库原理
  - 埃德加·科德(Edgar Codd) 1970年提出
- 埃德加·科德
  - IBM公司程序员
  - 1963年密歇根大学攻读博士学位
  - 1967年回到IBM做研究工作
  - 1970年在《美国计算机学会通讯》(CACM)上发表了关系数据库的开山之作“大型共享数据库数据的关系模型”(A Relational Model of Data for Large Shared Data Banks)
  - 1981年获得Turing award

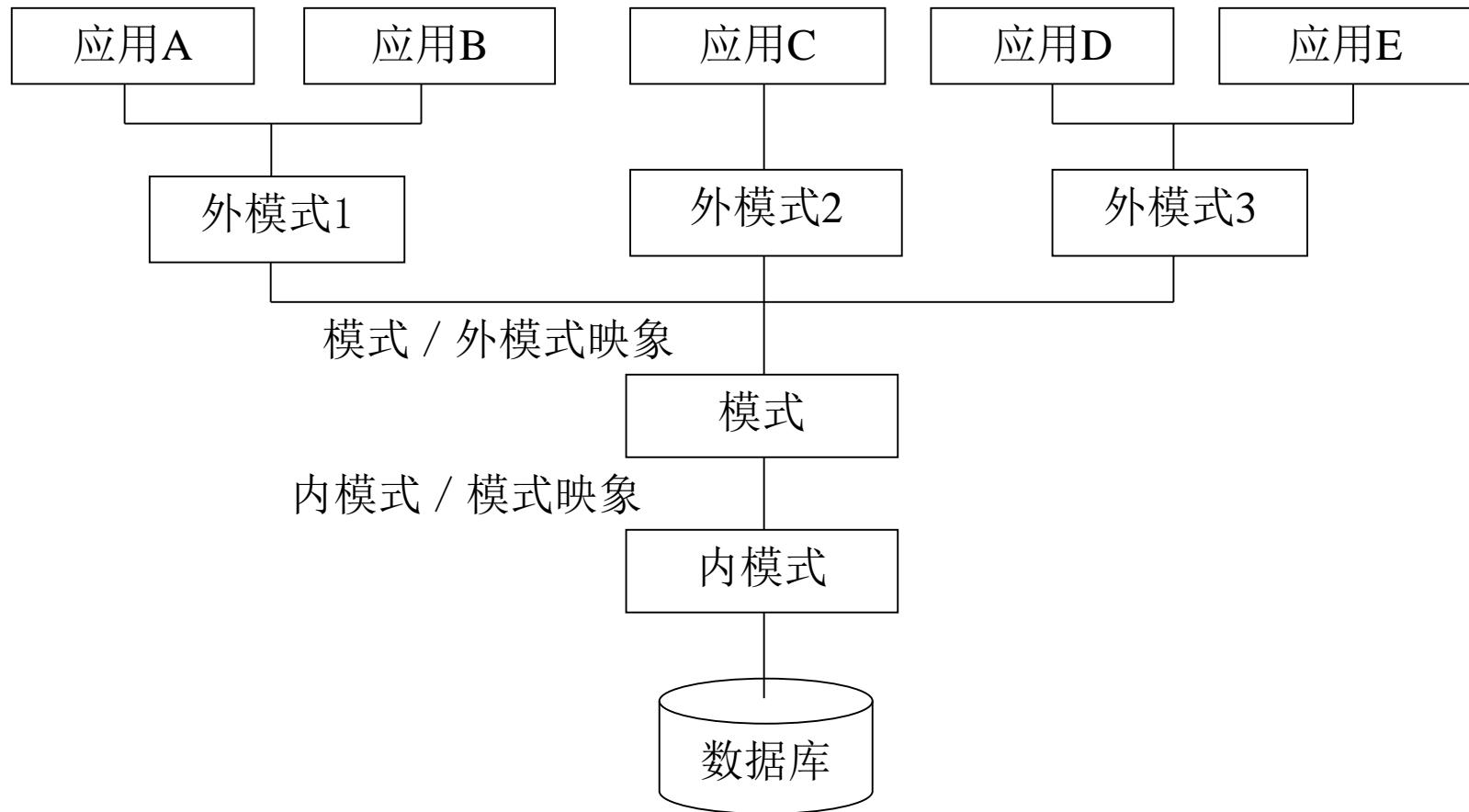


## 1.1.2 三级模式

---

- 模式(schema)定义或描述一个数据集合
- 模式 vs. 数据
  - 编程语言：变量类型 vs. 变量本身
- 模式相对稳定，反映数据结构及其联系
- 实例相对变动，反映数据库某一时刻的状态
  - 记录型 (学生姓名、性别、出生年月、籍贯、所在系别、入学时间)
  - 记录值 (陆鸣，男，2000，江苏，计算机系，2019)

# 关系数据库系统的三级模式结构



思考：三级模式结构与概念、逻辑和物理数据模型的差异？

# 模式

---

- 模式 (逻辑模式)
  - 用逻辑数据模型对数据库中全部数据的逻辑结构和特性的描述
  - 是数据库所有用户的公共数据视图
- 一个数据库只有一个模式
- 定义模式时，要定义：
  - 数据的逻辑结构 (数据项的名字、类型、取值范围等)
  - 数据之间的联系
  - 数据有关的安全性、完整性要求

# 外模式

---

- 外模式 (子模式或用户模式)
  - 对用户所用到的那部分数据的描述
  - 不同的用户因需求不同，看数据的方式可以不同，对数据的保密要求、使用的程序设计语言都可以不同，因此每个用户的外模式不一定相同
- 一个数据库可以有多个外模式
- 外模式是模式的一部分或是从模式推导而来的

# 内模式

---

- 内模式 (存储模式)
  - 用物理数据模型对数据的描述
  - 数据物理结构和存储方式的描述，是数据在数据库内部的表示方式，例如，是按B+树结构存储还是hash方法存储，是否压缩存储，是否建立索引，是否加密，如何进行存储管理等。
- 一个数据库只有一个内模式
- 内模式定义与修改是DBA (DataBase Administrators) 的责任

# 外模式 / 模式映象

---

- 定例外模式与模式之间的对应关系
- 每一个外模式都对应一个外模式 / 模式映象
- 映象定义通常包含在各自外模式的描述中
- 用途：保证数据的**逻辑独立性**
  - 当模式改变时，数据库管理员修改有关的外模式 / 模式映象，使外模式保持不变
  - 应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性

# 模式 / 内模式映象

---

- 模式 / 内模式映象定义了数据的逻辑结构与存储结构之间的对应关系
  - 例如，说明逻辑记录和字段在内部是如何表示的
- 数据库中模式 / 内模式映象是唯一的
- 该映象定义通常包含在模式描述中
- 用途：保证数据的**物理独立性**
  - 当数据库的存储结构改变了(例如，选用了另一种存储结构)，数据库管理员修改模式 / 内模式映象，使模式保持不变
  - 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性

# ANSI/SPARC Model

Users

Views describe how users see the data.



View 1

View 2

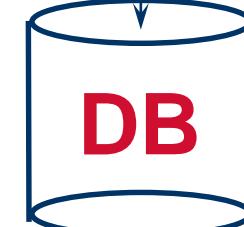
View 3

Conceptual schema defines logical structure

Conceptual Schema

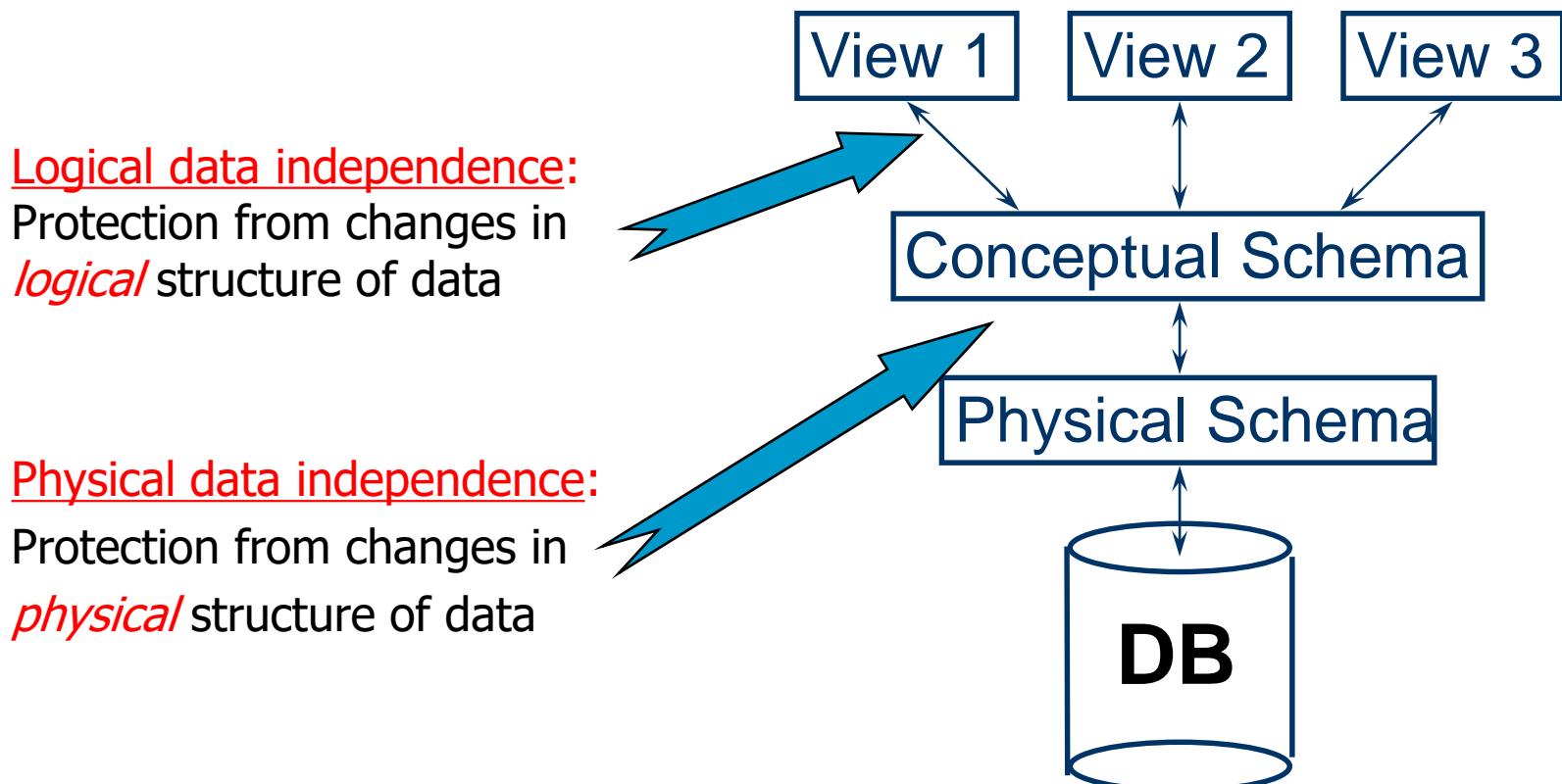
Physical schema describes the files and indexes used.

Physical Schema



# Data Independence: Two Flavors

- A Simple Idea: Applications should be insulated from how data is structured and stored



# 举例：大学数据库

---

- Conceptual schema

*Students*(*sid*: string, name: string, login: string, age: integer, gpa: real)

*Courses*(*cid*: string, cname: string, credits: integer)

*Enrolled*(*sid*: string, *cid*: string, grade: string)

- An Instance of Students Relation

| sid   | name  | login      | age | gpa |
|-------|-------|------------|-----|-----|
| 53666 | Jones | jones@cs   | 18  | 3.4 |
| 53688 | Smith | smith@eecs | 18  | 3.2 |
| 53650 | Smith | smith@math | 19  | 3.8 |

# 举例：大学数据库

---

- Conceptual schema (模式)

Students(sid: string, name: string, login: string, age: integer, gpa: real)

Courses(cid: string, cname: string, credits: integer)

Enrolled(sid: string, cid: string, grade: string)

- Physical schema (内模式)

- Relations stored as unordered files

- Index on first column of Students, first 2 cols of Enrolled

- External Schema (View, 外模式)

Course\_info(cid: string, enrollment: integer)

```
CREATE VIEW Course_info AS
```

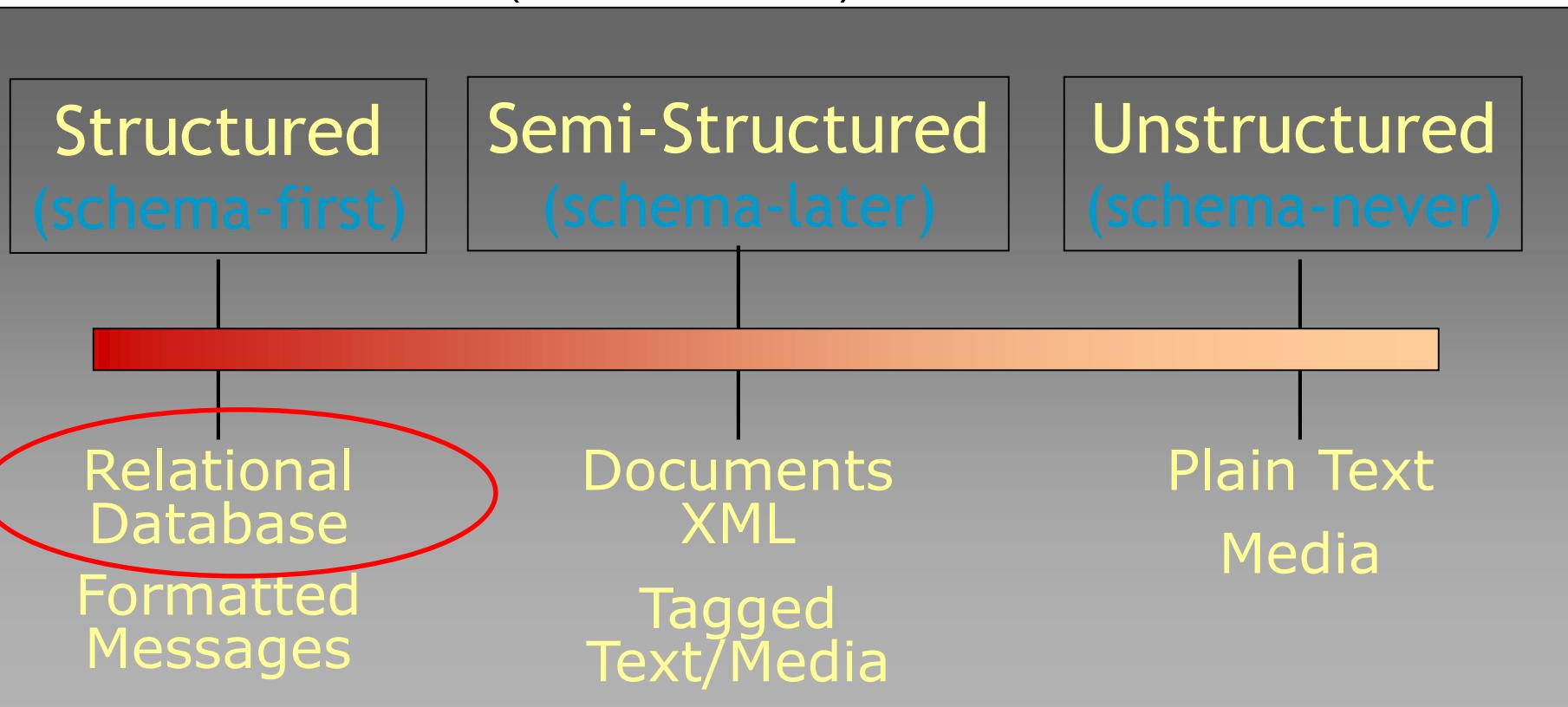
```
SELECT cid, Count (*) as enrollment
```

```
FROM Enrolled
```

```
GROUP BY cid
```

# 基于模式的数据分类

- 基于模式的数据分类：
  - 结构化数据 (Structured)
  - 半结构化数据 (Semi-Structured)
  - 非结构化数据 (Unstructured)





# DB-Engines Ranking (2021.9)

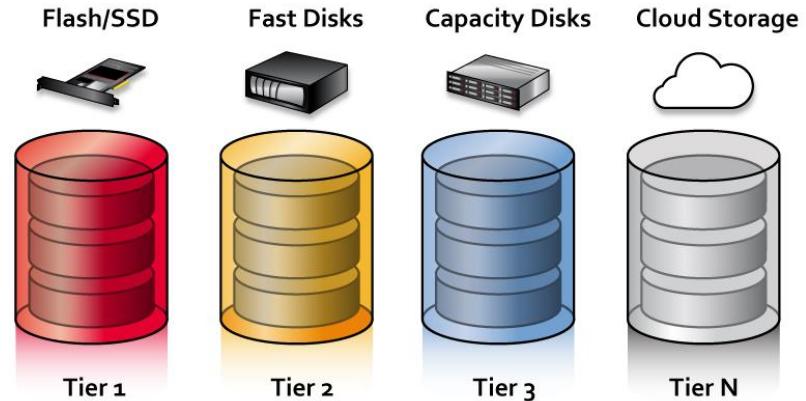
- Rank DBMSs according to their popularity

378 systems in ranking, September 2021

| Rank     |          |          | DBMS                 | Database Model             | Score    |          |          |
|----------|----------|----------|----------------------|----------------------------|----------|----------|----------|
| Sep 2021 | Aug 2021 | Sep 2020 |                      |                            | Sep 2021 | Aug 2021 | Sep 2020 |
| 1.       | 1.       | 1.       | Oracle               | Relational, Multi-model    | 1271.55  | +2.29    | -97.82   |
| 2.       | 2.       | 2.       | MySQL                | Relational, Multi-model    | 1212.52  | -25.69   | -51.72   |
| 3.       | 3.       | 3.       | Microsoft SQL Server | Relational, Multi-model    | 970.85   | -2.50    | -91.91   |
| 4.       | 4.       | 4.       | PostgreSQL           | Relational, Multi-model    | 577.50   | +0.45    | +35.22   |
| 5.       | 5.       | 5.       | MongoDB              | Document, Multi-model      | 496.50   | -0.04    | +50.02   |
| 6.       | 6.       | ↑ 7.     | Redis                | Key-value, Multi-model     | 171.94   | +2.05    | +20.08   |
| 7.       | 7.       | ↓ 6.     | IBM Db2              | Relational, Multi-model    | 166.56   | +1.09    | +5.32    |
| 8.       | 8.       | 8.       | Elasticsearch        | Search engine, Multi-model | 160.24   | +3.16    | +9.74    |
| 9.       | 9.       | 9.       | SQLite               | Relational                 | 128.65   | -1.16    | +1.98    |
| 10.      | ↑ 11.    | 10.      | Cassandra            | Wide column                | 118.99   | +5.33    | -0.18    |

# Current Market

- Relational DBMSs anchor the software industry
  - Elephants: Oracle, IBM, Microsoft, Teradata, HP, EMC, ...
  - Open source: MySQL, PostgreSQL
  - New “Big Data” Entrants: Hive & Pig (Hadoop), Shark (Spark),
- Obviously, Search
  - Google & Bing
- Open Source “NoSQL”
  - Hadoop MapReduce, Spark
  - Key-value stores: Cassandra, Riak, Voldemort, Mongo, ...
- Cloud services
  - Amazon, Google AppEngine,
  - MS Azure, Heroku, ...



# 第一章 地理空间数据库概论

---

- 1.1 关系数据库基本概念
- 1.2 空间数据库基本概念
  - 1.2.1 空间数据
  - 1.2.2 空间数据的特征
  - 1.2.3 空间数据库
  - 1.2.3 空间数据库系统
- 1.3 空间数据管理技术的产生与发展
- 1.4 现有空间数据库标准简介
- 1.5 现有空间数据库管理系统产品简介 (自学)

# 1.2 基本概念

---

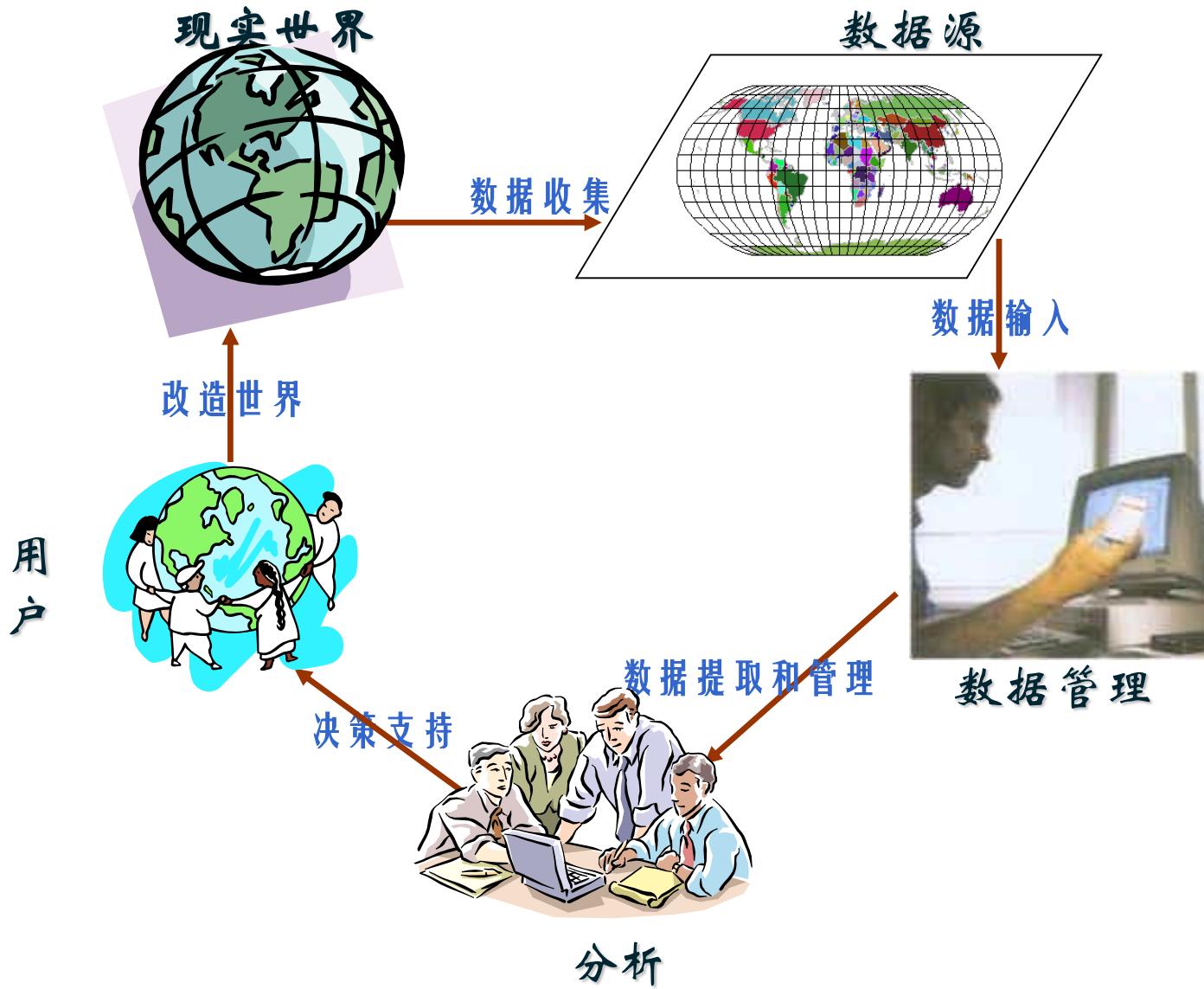
- 空间数据库
  - 在地球表面某一范围内与空间地理相关、反映某一主题信息的数据集合
  - 以**空间目标**作为存储对象的专业数据库
  - GIS核心和基础
- 空间数据库应用
  - 土地利用、资源管理、环境监测、交通运输、城市规划等
- 空间数据库衡量指标
  - 规模大小、响应速度、共享程度

# 1.2 基本概念

---

- GIS is a software to visualize and analyze spatial data using spatial analysis functions such as
  - Search: Thematic search, search by region, (re-)classification
  - Location analysis: Buffer, corridor, overlay
  - Terrain analysis: Slope/aspect, catchment, drainage network
  - Flow analysis: Connectivity, shortest path
  - Distribution: Change detection, proximity, nearest neighbor
  - Spatial analysis/Statistics: Pattern, centrality, autocorrelation, indices of similarity, topology: hole description
  - Measurements: Distance, perimeter, shape, adjacency, direction
- GIS uses SDBMS to store, search, query, share large spatial data sets

# 1.2 基本概念



## 1.2.1 空间数据

---

- 空间数据
  - 以地球表面空间位置为参照的自然、社会和人文经济景观数据
- 广义上包括
  - 文字、数字、图形、影像、声音、图像等多种表现形式
  - 如地名地址、数字高程、矢量地图、遥感影像、地理编码数据、多媒体地图
- 空间数据分为**矢量数据**和**栅格数据**

思考：哪些空间数据分别属于自然、社会和人文经济景观数据？

## 1.2.1 空间数据

---

- 矢量数据
  - 矢量数据是一种用点、线、面等基本空间要素来表示人们赖以生存的自然世界的数据
- 不可再分的最小单元现象称为**空间实体**
  - 对存在于这个自然世界中地理实体的抽象
  - 包括点、线、多边形等基本类型

## 1.2.1 空间数据

---

- 空间实体举例
  - 一根电线杆 → 点
    - 所处的位置信息，电线杆高度及其他相关信息
  - 一条道路 → 线
    - 道路长度、宽度、起点、终点及道路等级等相关信息
  - 一个湖泊 → 多边形
    - 湖泊的周长、面积和水质等信息

## 1.2.1 空间数据

- 空间实体举例
  - 一根电线杆 → 点
  - 一条道路 → 线
  - 一个湖泊 → 多边形

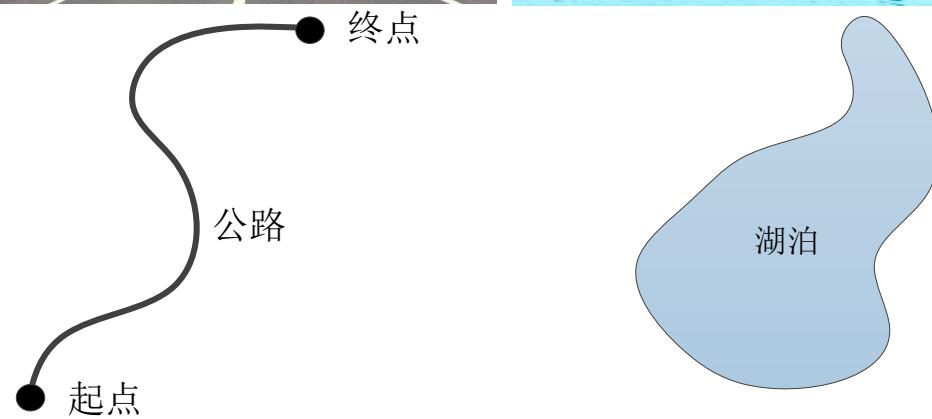
现实世界



矢量表达

● 电线杆2

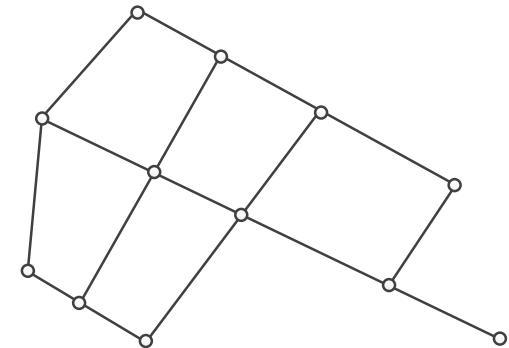
● 电线杆3



思考：图中存在哪些实体的空间关联和时空关系？

## 1.2.1 空间数据

- 矢量数据
  - 空间实体本身的空间位置及属性信息
  - 空间实体相互之间的关系 – 空间关系
- 空间关系 → 拓扑关系 (**topology**)
  - 表示点、线、多边形等实体之间的空间联系
- 例如
  - 网络结点与网络之间的枢纽关系
  - 边界线与多边形实体间的构成关系
  - 多边形实体与岛或内部点的包含关系



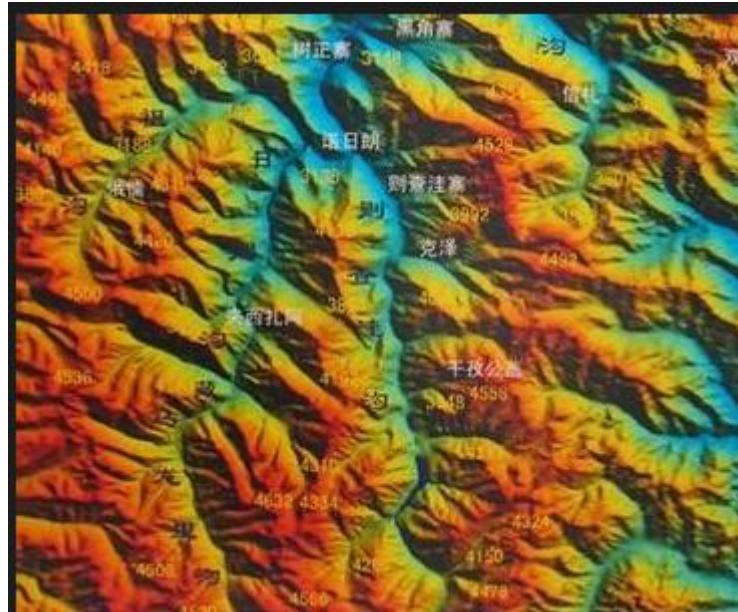
## 1.2.1 空间数据

---

- 棚格数据
  - 把地理空间中的事物和现象作为**连续**的变量或体看待
  - 如大气污染植被覆盖、土壤类型、地表温度等
- 将地面划分为均匀的网格，每个网格作为一个像元，像元的位置由所在列和行号确定，像元所含有的代码表示其属性类型或仅是与其属性记录相联系的指针

## 1.2.1 空间数据

- 栅格数据



数字高程模型 (DEM) 数据



影像数据

## 1.2.2 空间数据的特征

---

- 由于空间数据的复杂性和特殊性，一般商用数据库管理系统难以满足其管理需求。空间数据主要具有以下重要特征
  - 空间特征
  - 非结构化特征
  - 空间关系特征
  - 时态特征
  - 多尺度特征

思考：上述特征与空间计算（空间关联和时空查询）的对应关系

## 1.2.2 空间数据的特征

---

- 空间特征
  - 每个空间对象都具有空间坐标，即空间对象隐含了空间分布特征
  - 空间数据组织需要考虑它的空间分布特征
  - 除了属性索引外，还需要建立空间索引

## 1.2.2 空间数据的特征

---

- 非结构化特征
  - 关系数据库中数据记录是**结构化的**
  - 结构化数据
    - 满足关系模式的范式基本要求，可以用二维表结构来逻辑表达的数据
  - 非结构化的数据
    - 不方便用数据库二维逻辑表来表现的数据，包括文本、图片、**XML**、**HTML**、音频、视频等

## 1.2.2 空间数据的特征

---

- 非结构化特征
  - 空间数据是一种**非结构化**数据
    - 空间实体是**不定长**的，例如一条弧段可能包含两对坐标点，也有可能10万对坐标点
    - 空间实体是**非原子**的，有的甚至是**嵌套**的，例如一个多边形可能包含多条弧段
  - 通用的关系数据库管理系统难以直接管理空间数据

## 1.2.2 空间数据的特征

---

- 空间关系特征
  - 空间数据包括空间坐标和拓扑关系
    - 方便空间数据的查询和空间分析 (几何对象模型和空间网络模型)
    - 给空间数据的一致性和完整性维护增加了复杂性
      - 特别是一些没有直接记录空间坐标信息的几何对象 (例如拓扑的面状表面仅记录组成它的弧段的标识), 在进行相关的查找、显示和分析操作时, 都要操纵和检索多个数据文件才行完成

## 1.2.2 空间数据的特征

- 时态特征
  - 反映地理实体的状态和演变过程的重要组成部分
  - 如何组织、管理地理实体随时间变化信息(或时空信息),是空间数据库面临的新课题
  - 现有的空间数据库基本不具有管理空间数据的时间动态性,只是描述数据的**瞬时状态**
    - 如果数据发生变化,新数据将代替旧数据,即成为另一个瞬时状态,就数据将会消失,无法对数据的更新变化进行分析,更不能预测未来的趋势
    - 地籍变更、环境监测、抢险救灾、交通管理等都需要时空信息

思考：时态特性对时空查询的影响，如查询出租车速度评估道路拥堵程度？

## 1.2.2 空间数据的特征

---

- 多尺度特征
  - 地球系统是各种不同级别子系统组成的复杂巨系统，各个级别的子系统在空间规律和事件长短方面存在很大差异，而其由于空间认知水平、认知精度和比例尺等不同，地理实体的表现形式也不同
  - **空间多尺度**是指根据地学过程或地理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**是指地学过程或地理特征具有一定的自然节律性，其时间周期长短不一

思考：多尺度特征对时空数据建模的影响

## 1.2.3 空间数据库

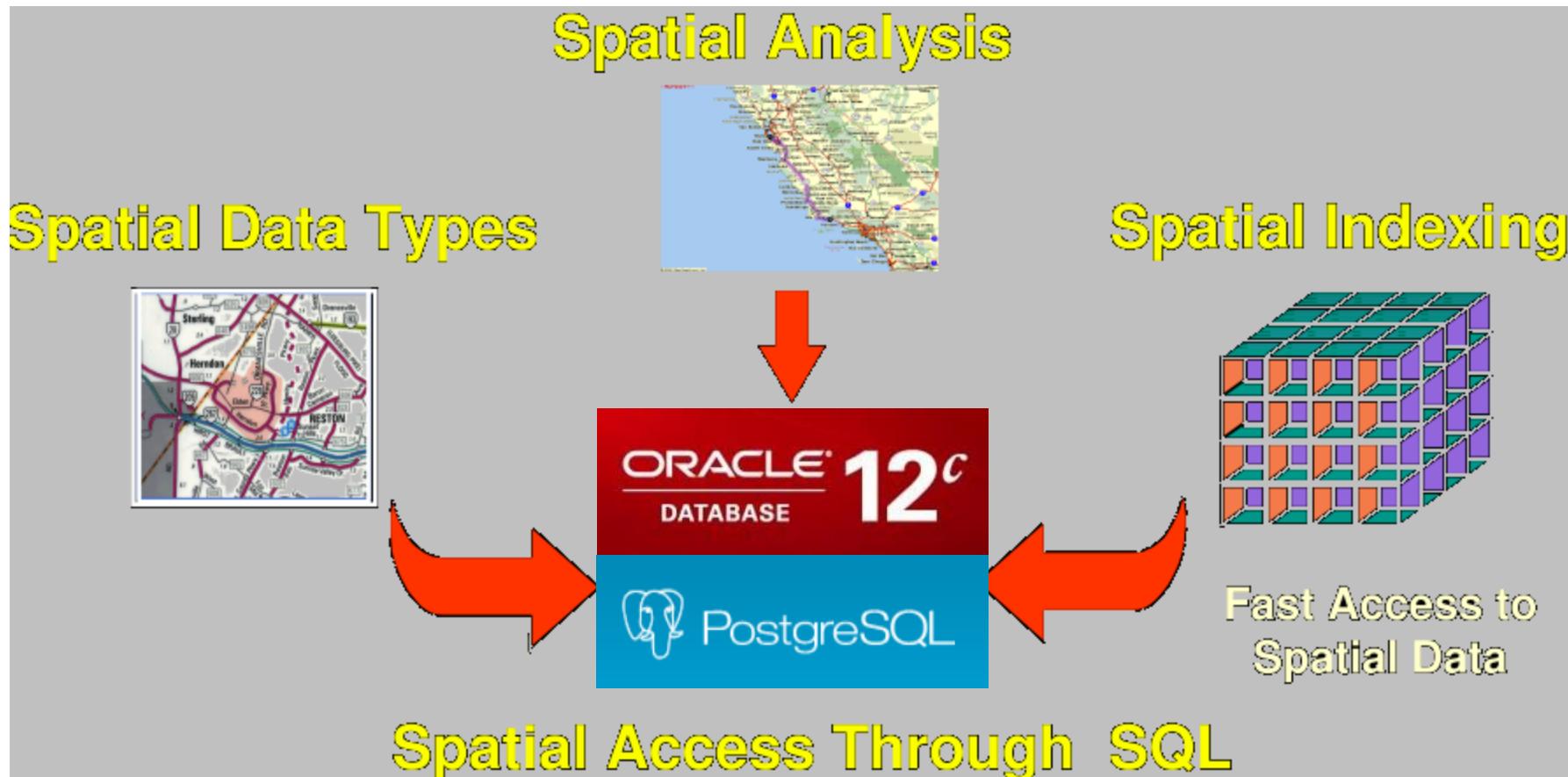
- 尚无公认统一的定义
  - 在地球表面某一范围内与空间地理相关的反应某一主题的数据集合
- 要求
  - 按一定的**数据模型**组织、描述和存储，具有较小的**冗余度**、较高的数据**独立性和易扩展性**，并可为各种用户**共享**
  - 例如国家基础地理信息数据库、资源环境数据库

地理空间数据库 = 地理空间数据 + 数据库管理系统



## 1.2.3 空间数据库

- 空间数据库三大要素



# 1.2.3 空间数据库

## ● 三层架构

Spatial Application

GIS

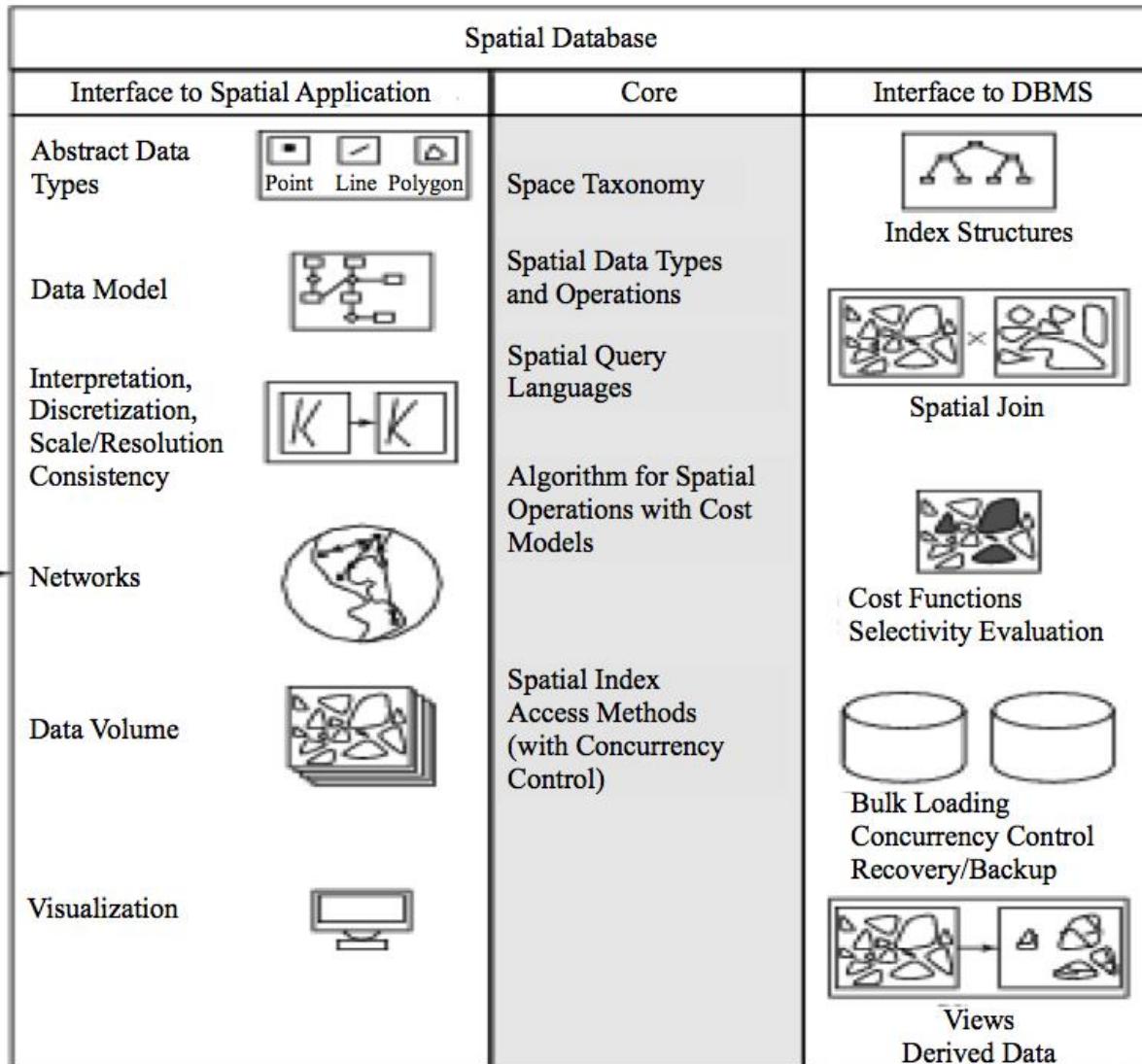
MMIS

CAD

DBMS



Object-  
Relational  
Database  
Servers



## 1.2.3 空间数据库

---

- 空间数据库与一般数据库相比，具有以下特点
  - 数据量大
    - 一个城市达几十G，影像达几百G
    - 在二维空间上划分块或图幅，在垂直方向上划分层来进行组织
  - 空间数据与属性数据的集合
  - 应用广泛
    - 自然、经济、社会等信息的80%与地理空间位置相关
    - 广泛应用于地理研究、环境保护、国土资源管理、资源开发、市政管理、交通管理等领域

## 1.2.4 空间数据库管理系统

---

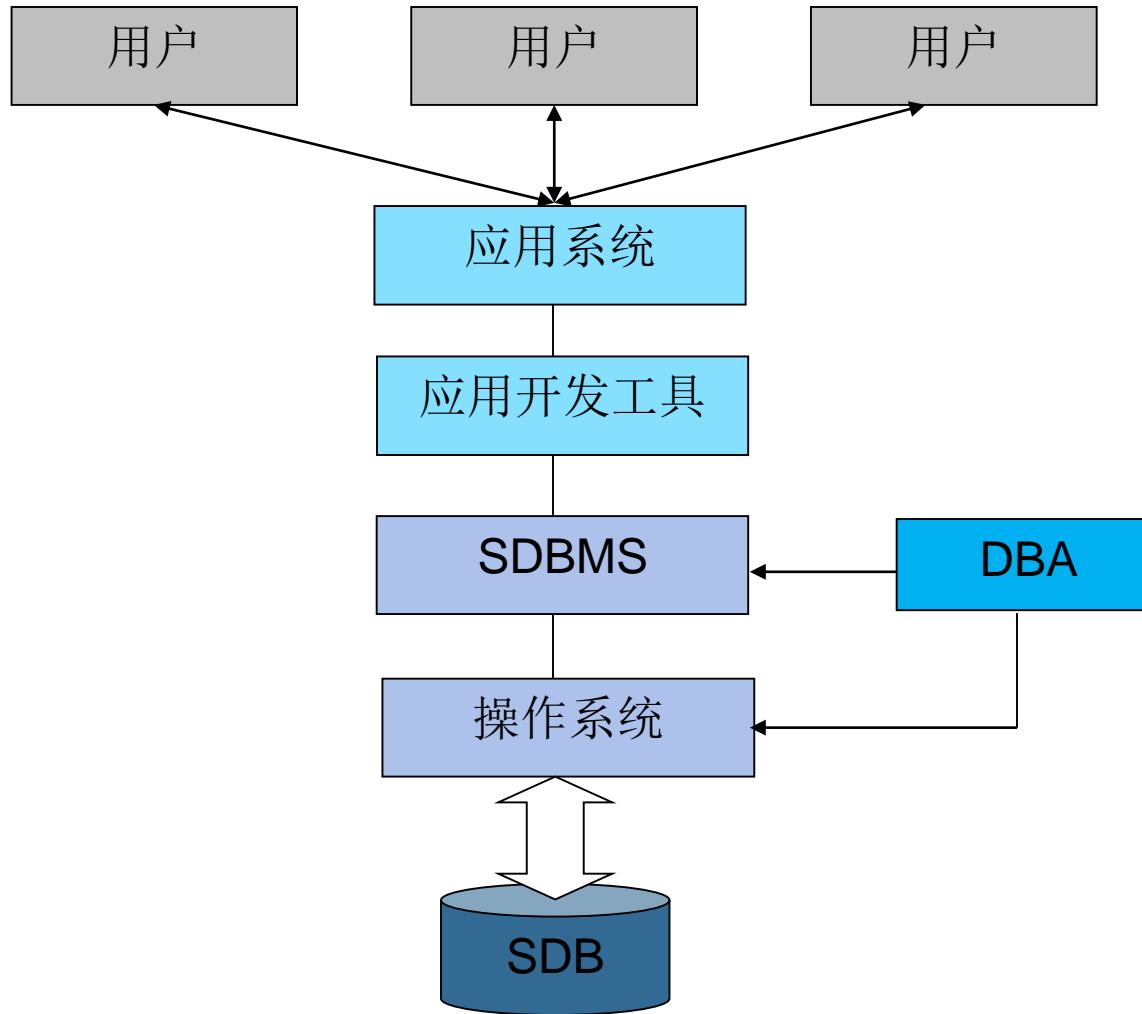
- 空间数据库管理系统 (Spatial Database Management System, SDBMS)
  - Oracle Spatial
  - PostgreSQL PostGIS
  - DB2 Spatial Extender
- 主要功能
  - 空间数据的定义与操纵 (SDDL, SDML)
  - 空间数据的组织、存储和管理 (存取效率)
  - 后台的事务管理和运行管理 (DB系统管理员)
  - 数据库的建立与维护

## 1.2.4 空间数据库系统

---

- 空间数据库系统 (Spatial Database System, SDS)
  - 由空间数据库及其管理软件、应用软件组成
  - 存储介质、处理对象和管理系统的集合体
- 组成部分
  - 空间数据库
  - 空间数据库管理系统
  - 数据库管理员
  - 用户和应用程序

## 1.2.4 空间数据库系统



## 1.2.4 空间数据库系统

---

- 数据库系统中有多种用户，他们分别扮演着不同的角色，承担不同的任务。开发、管理和使用数据库系统的人员主要
  - 数据库管理员 (DBA)
  - 系统分析员和数据库设计人员
  - 应用程序员
  - 最终用户

## 1.2.4 空间数据库系统

---

- 数据库管理员(DBA)具体的职责包括
  - 决定数据库中的信息内容和结构
  - 决定数据库的存储结构和存取策略
  - 定义数据的安全性要求和完整性约束条件
  - 监控数据库的使用和运行
  - 数据库的改进和重组重构

## 1.2.4 空间数据库系统

---

- 系统分析员负责应用系统的需求分析和规范说明，他们要和用户及DBA相结合，确定系统的硬软件配置并参与数据库系统的概要设计
- 数据库设计人员负责数据库中数据的确定、数据库各级模式的设计。数据库设计人员必须参加用户需求调查和系统分析，然后进行数据库设计。在很多情况下，数据库设计人员就由数据库管理员担任

## 1.2.4 空间数据库系统

---

- 应用程序员负责设计和编写应用系统的程序模块，并进行调试和安装
- 用户是指最终用户(End User)，可以分为三类
  - 偶然用户：不经常访问数据库。每次访问数据库时往往需要不同的数据库信息，一般是企业或组织机构的高级管理人员
  - 简单用户：主要工作是查询和更新数据库，一般都是通过应用程序员精心设计并具有友好界面的应用程序存取数据库
  - 高级用户：一般都比较熟悉数据库管理系统的各种功能，能够直接使用数据库语言访问数据库，甚至能够基于数据库管理系统的API编制自己的应用程序

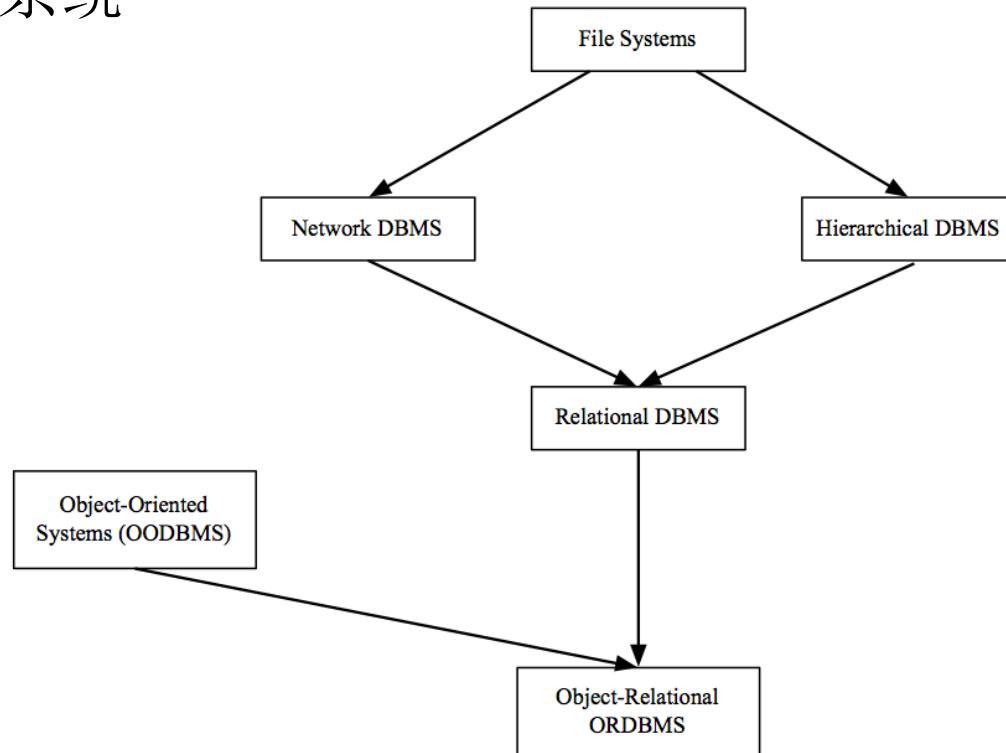
# 第一章 地理空间数据库概论

---

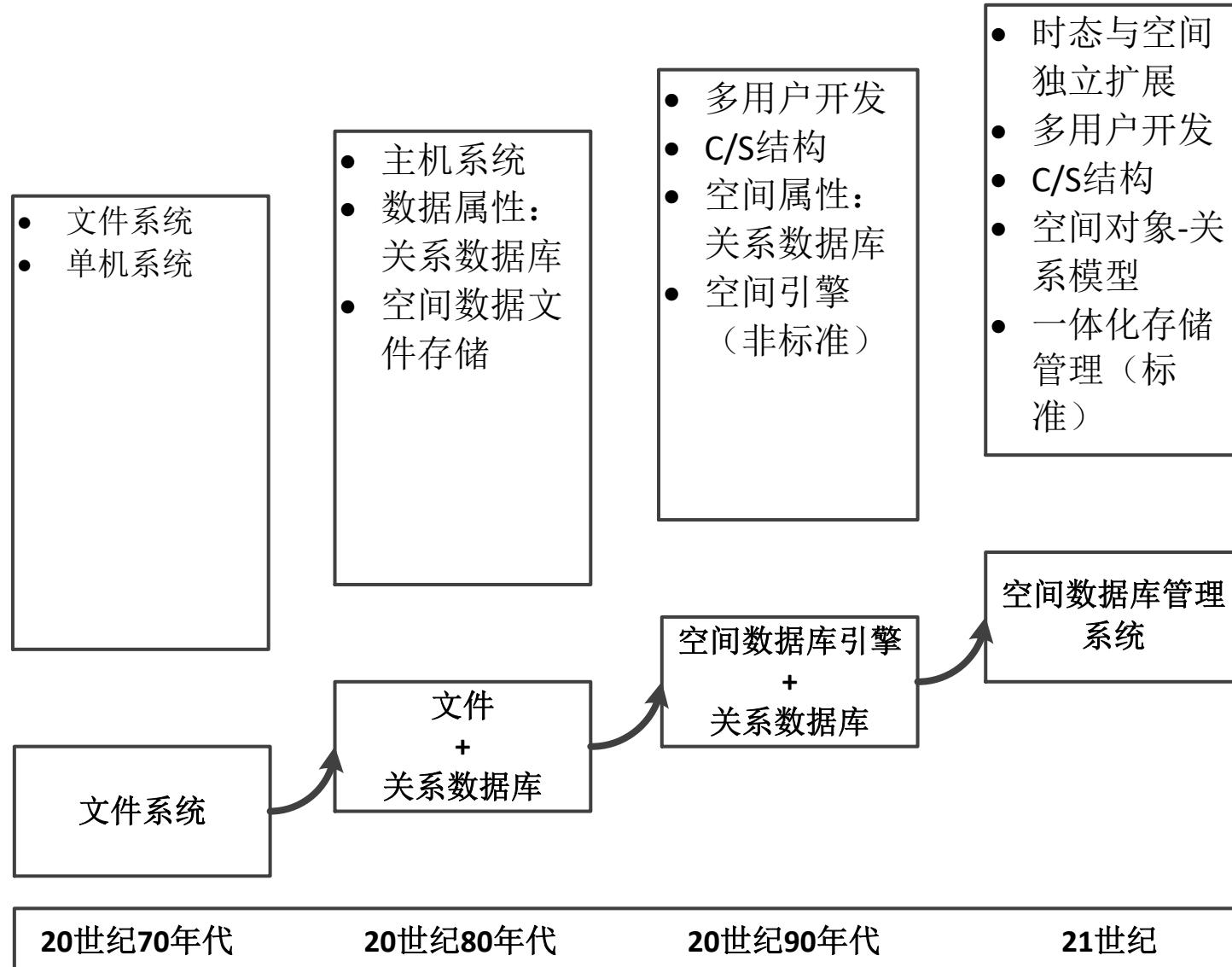
- 1.1 关系数据库基本概念
- 1.2 空间数据库基本概念
- 1.3 空间数据管理技术的产生与发展
  - 1.3.1 文件系统
  - 1.3.2 文件与关系数据库混合管理系统
  - 1.3.3 空间数据引擎
  - 1.3.4 对象关系型数据库管理系统
- 1.4 现有空间数据库标准简介
- 1.5 现有空间数据库管理系统产品简介 (自学)

# 1.3 空间数据管理技术的产生与发展

- 发展与演变
  - 文件系统
  - 文件关系混合系统
  - 空间数据库引擎
  - 对象关系型数据库管理系统



# 1.3 空间数据管理技术的产生与发展



## 1.3.1 文件系统

- 加拿大政府从20世纪60年代中期开始，历经10年时间，研发了世界上第一个地理信息系统——加拿大地理信息系统 (CGIS)
- 20世纪50年代-70年代
- 第一代空间应用系统



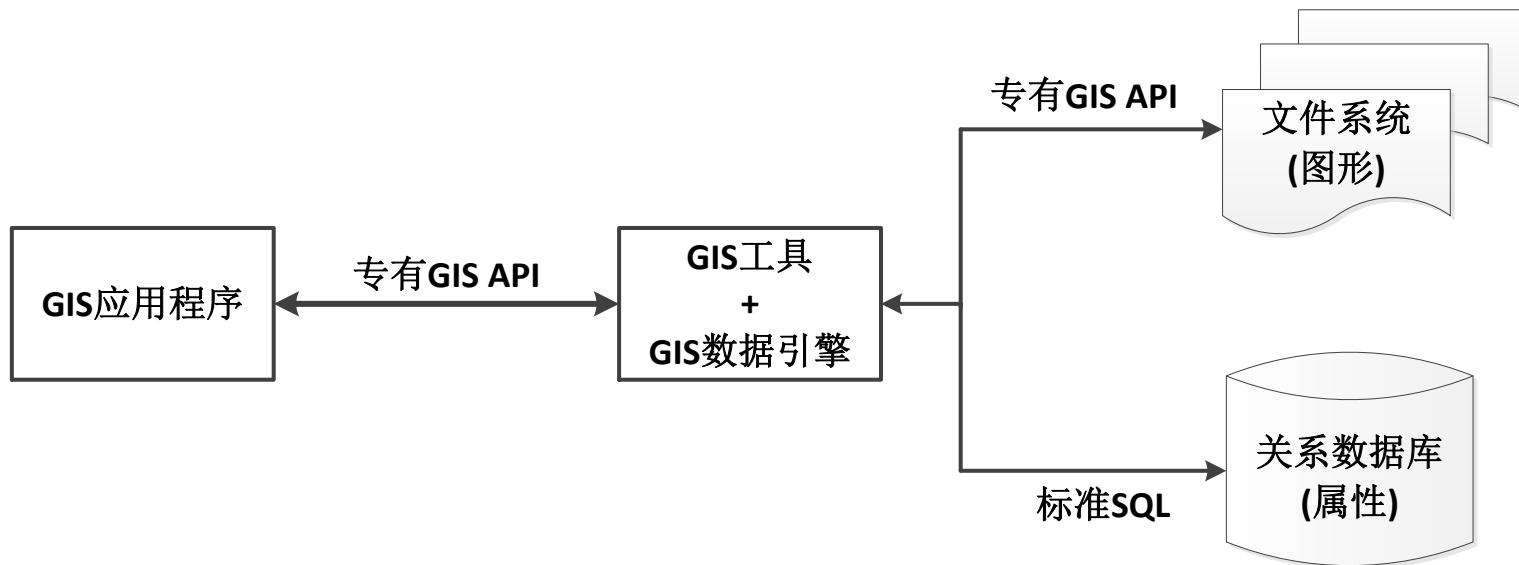
## 1.3.2 文件与关系数据库混合管理系统

---

- 20世纪80年代，关系数据库发展并成熟
  - 点、线、面用关系数据库进行存储
  - 非结构化常常导致效率低下，不利于管理和共享
- 混合管理系统
  - 文件系统管理几何图形数据
  - 关系数据库管理属性数据
  - 他们之间的联系通过目标标识或内部连接码进行
    - 对象唯一标识符OID
  - 第二代空间应用系统

## 1.3.2 文件与关系数据库混合管理系统

- 文件管理系统功能较弱
  - 数据安全性、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



### 1.3.3 空间数据引擎

- 关系数据库支持可变长文本字符的大二进制 (binary large object, BLOB)字段
- 1996年美国环境系统研究所与Oracle合作，开发空间数据引擎 (Spatial Database Engine, SDE，后更名为ArcSDE)
  - 图形坐标数据作为一个**二进制数据类型**，由数据库管理系统进行存储
  - SDE提供一组空间数据的操作函数，完成空间数据的转换，以及数据的索引调度和空间数据的存储管理
- 空间数据引擎
  - ESRI的ArcSDE、SuperMap的SDX、中地的MapGIS SDE、开源的TerraLib
- 第三代空间应用系统

### 1.3.3 空间数据引擎

- 空间数据引擎



- 中间件解决方案

- 独立于数据库内核，难以利用成熟的数据管理、访问技术，不支持空间结构化查询语言 (SSQL)
- 不同厂商数据格式定义不同

## 1.3.4 对象关系型数据库管理系统

---

- 对象关系型数据库管理系统 (Object-relational database management system, ORDBMS)
  - 支持SQL，具有良好的通用性
  - 具有面向对象特性，支持复杂对象及其行为
- 能够直接存储和管理非结构化的空间数据
  - Oracle Spatial
  - IBM的DB2 Spatial Extender
  - 微软的SQL Server Spatial
  - 开源的PostGIS

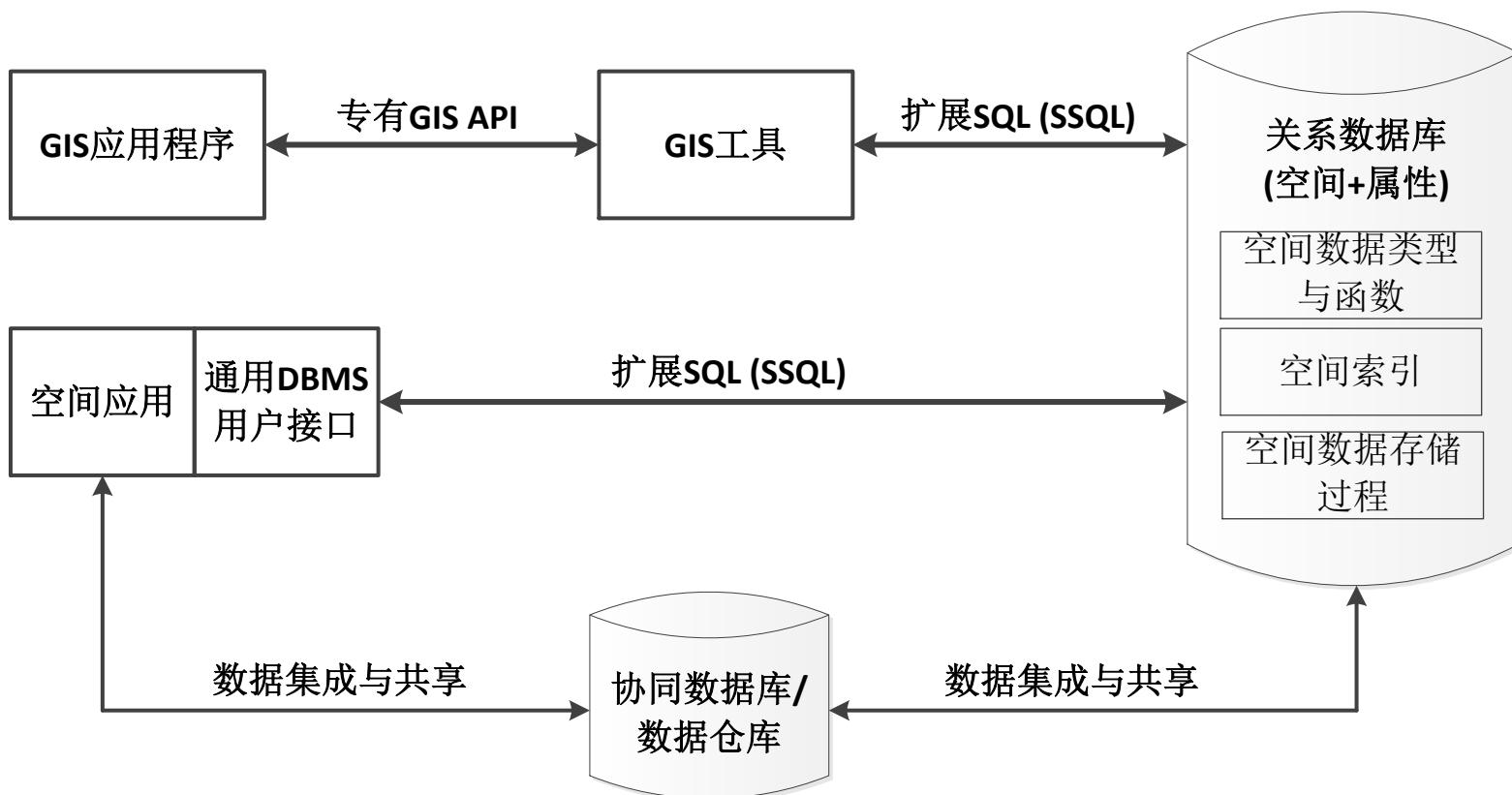
## 1.3.4 对象关系型数据库管理系统

---

- ORDBMS提供
  - 类、继承
  - 用户自定义类型、函数、索引和规则
- 对各种空间对象、操作函数及其索引进行预先定义，形成不同的空间数据类型，支持空间数据的存储、管理和分析
- 第四代空间应用系统

# 1.3.4 对象关系型数据库管理系统

- 对象关系型数据库管理系统



# 后两种空间数据管理方案对比与分析

| 项目   | 空间数据引擎（寄生模式）                              | 对象关系空间数据库（融合模式）  |
|------|---|--|
| 技术特点 | 中间件技术                                     | 数据库技术  |
| 代表产品 | ArcSDE、SuperMap SDK+、MapGIS TerraLib (开源) | Oracle Spatial 、 DB2 Spatial Extender、PostGIS (开源)   |
| 对比分析 | 优点  | <ul style="list-style-type: none"><li>支持通用<b>RDBMS</b>，可跨数据平台</li><li>与特定<b>GIS</b>平台结合紧密，有较高的空间处理效率</li></ul>     |
|      | 缺点  | <ul style="list-style-type: none"><li>难以利用<b>DBMS</b>的内核技术</li><li>难以支持扩展<b>SQL</b></li><li>难以实现数据共享与互操作</li></ul> |

# 第一章 地理空间数据库概论

---

- 1.1 关系数据库基本概念
- 1.2 空间数据库基本概念
- 1.3 空间数据管理技术的产生与发展
- 1.4 现有空间数据库标准简介
  - 1.4.1 SFA SQL
  - 1.4.2 SQL/MM
  - 1.4.3 小结
- 1.5 现有空间数据库管理系统产品简介 (自学)

# 1.4 现有空间数据标准简介

---

- 开放地理空间信息协会
  - Open Geospatial Consortium, OGC
  - 地理信息简单要素的SQL实现规范
    - Simple Feature Access SQL, SFA SQL
- 国际标准化组织/国际电工委员会第一联合技术委员会/数据管理和交换分技术委员会 (ISO/IEC JTC1 SC32)
  - SQL多媒体及应用包的第三部分 (SQL Multimedia Part3: Spatial, SQL/MM)

## 1.4.1 SFA SQL

---

- OGC于1994年成立
- 主要任务
  - 研制公众可用的开放式地理信息规范(*open geographic information specifications, OGIS*), 使其具有网络环境中透明地共享异构地理数据及其处理资源的能力
- SFA SQL
  - 1999年提出
  - 说明了简单地理要素(点, 线, 多边形等)的对象模型及其发布、存储、读取操作的接口标准

## 1.4.1 SFA SQL

---

- SFA SQL
  - 2005年进一步细化了相关内容，添加了注记文字(annotation text)，将其修订为简单要素访问规范(simple feature access, SFA)1.1.0版
  - 第一部分(Part1 Common Architecture)定义几何对象的通用架构，描述了通用的简单要素地理集合对象模型，及集合对象的不同表达方式和空间参考系统的表达方式，具有平台独立性
  - 第二部分(Part2: SQL Option, SFA SQL)定义了第一部分定义的简单要素模型在数据库中的实现，给出了内模式下几何类型(geometry type)的定义及相关实现
  - 2006年10月，推出了SFA 1.2.0版，目前该实现规范已被ISO TC211吸纳为ISO19125系列标准

## 1.4.2 SQL/MM

---

- ISO/IEC JTC1于1987年成立
- 主要任务
  - 指定国际标准和技术报告，涉及内容包括系统和工具的规范、设计和开发，涉及信息的采集、表示、处理、安全、传送、交换、显示、管理、组织、存储和检索等
- SQL/MM第三部分空间定义了矢量数据存储与检索的相关标准，解释了基于这些数据类型如何使用存储、获取和处理空间数据
- 已推出第三版

## 1.4.3 小结

---

- 这两个标准公共部分的接口已经相互兼容，但在内容覆盖面和某些概念的界定上存在一定的差异
  - SFA SQL在标记文本类型、空间数据存储实现上比SQL/MM定义的更宽泛
  - SQL/MM涉及了SFA SQL尚未涉及的拓扑数据结构、网络模型等方面的内容
- 没有统一的**SDB**标准，导致**SDBMS**差异
  - PostGIS更符合SFA SQL标准
  - Oracle Spatial更兼容SQL/MM标准

# 第一章 地理空间数据库概论

---

- 1.1 关系数据库基本概念
- 1.2 空间数据库基本概念
- 1.3 空间数据管理技术的产生与发展
- 1.4 现有空间数据库标准简介
- 1.5 现有空间数据库管理系统产品简介 (自学)

# 1.5 现有空间数据库管理系统产品简介

---

- Oracle Spatial
  - 7.2引入了内嵌式空间扩展技术——MultiDimension (MD)
  - 8该产品名称被修改为**Spatial Data Cartridge**和**Spatial Data Option** (空间几何对象的坐标串主要依靠关联表来存储，管理效率低下)
  - 8i-10g使用新的**SDO\_GEOMETRY**数据类型存储空间数据
  - 9i开始使用**SRID**属性提供对集合参考坐标系的支持

# 1.5 现有空间数据库管理系统产品简介

- Oracle Spatial

- 10g引入很多高级的模型与功能，如**EPSG**坐标系模型、网络数据模型(**NetWork**)等，空间选项增加了很多新特征，如**3D**几何对象，**Web**服务
- Oracle Spatial提供的函数集
  - 一种描述几何数据存储、语法、语义的模式**MDSYS**
  - 一组空间索引机制
  - 一组实现感兴趣区域查询和空间联合查询的算子和函数
  - 一组处理结点、边和表的拓扑数据模型
  - 一个存储、检索、查询、分析栅格数据的工具包(**GeoRaster**)
  - 一个网络数据模型

- 12c Spatial and Graph



# 1.5 现有空间数据库管理系统产品简介

---

- DB2 Spatial Extender

- IBM和ESRI于1998年在DataJoiner基础上联合开发
- 7.1版提供DB2 Spatial Extender
- 支持空间数据与传统数据在存储、管理和修改上的整合，并可以基于空间信息及其属性的结构来扩展已有的数据类型
- 在数据库中提供图形信息系统能力，使用户能够发现和开发他们数据库的空间职能

# 1.5 现有空间数据库管理系统产品简介

---

- SQL Server Spatial 2008
  - 2008年提供了对空间数据无缝的支持和整合，支持空间数据标准
  - 整合了地理坐标系和平面坐标系数据类型以及针对该类型的相关操作，针对新的空间数据类型提供存储新的操作分析能力，提供针对多级网格索引结构来加速查询检索的性能
  - 为了直观的展示空间数据，可以将查询结果使用管理控制台和相关的前端工具直观地加以显示

# 1.5 现有空间数据库管理系统产品简介

---

- PostGIS

- PostgreSQL的空间扩展，功能类似于Oracle Spatial Cartridge和IBM DB2的Spatial Extender
- 由Refractions Research公司开发，基于GNU GPL开源
- 新的空间数据类型、函数以及运算符使用C代码加以实现并使用SQL语句加以注册，PostgreSQL服务器将C的动态库编译并动态加载

# 1.5 现有空间数据库管理系统产品简介

---

- PostGIS

- 使用R树索引作为空间索引，其基于GIST(generalized search tree)索引模式实现
- 松耦合模块，因为空间查询处理算法使用用户级的API加以实现，而在数据库内核与存储系统级别上支持R树索引
- 只支持元组级的嵌套循环连接
- OGC标准规范的最佳实现

# 1.5 现有空间数据库管理系统产品简介

---

- MySQL Spatial
  - 4.0加入了**Spatial**扩展功能，实现了**OpenGIS**规定的集合数据类型，支持简单空间运算
  - 但到现在一直没有更新和增强
  - 早先**MySQL**在**SQL**上对空间运算支持的不完善，只支持基于最小边界矩形的关系判断，所以**MySQL**是开源数据库中一个不太满意的选择

# 1.5 现有空间数据库管理系产品简介

| 项目   | Oracle Spatial  | DB2 Spatial Extender                                    | SQL Server Spatial 2008  | MySQL Spatial              | Post GIS   |
|------|---|---|--------------------------|----------------------------|--|
| 数据类型 | 点, 线, 多边形<br>实体集, 多点, 多线, 实体集<br>多多边形                                 | 点, 线, 多边形<br>实体集  | 平面数据, 测量数据               | OpenGis几何类型、WKB、WKT        | 点, 线, 多边形<br>实体集, 多点, 多线, 多多边形, WKB, EWKB, EWKT                                |
| 遵循规范 | SQL/MM, SFA SQL   | ISO SQL/MM, GML, SFA SQL                                | ISO SQL/MM, GML, SFA SQL | SFA SQL                    | ISO SQL/MM, SFA SQL扩展数据类型  |
| 存储模式 | 扩展数据类型  | 通过类型层次组织  | 系统数据类型(原生)               |                            |  |
| 空间索引 | R树, Quadtree  | 提供基于网格的三层空间索引, 该索引技术是基于传统的分层B树索引形成的, 与ArcSDE的优化网格空间索引类似 | 4级网格索引                   | R树(MyISAM <sup>f</sup> )   | GiST(R树)   |
| 空间操作 | 空间操作函数<br>聚合函数<br>坐标系统转化<br>地理编码<br>几何操作<br>线性参照系统<br>数据迁移<br>空间分析与挖掘 | 数据格式转化函数<br>空间比较函数<br>几何对象属性函数<br>创建新几何体函数<br>其它        | 约70种函数和方法                | 对象创建<br>对象关系分析<br>数据转换(部分) | 对象创建<br>空间测量<br>几何对象关系分析<br>数据转换<br>数据读写<br>对象编译<br>多位坐标空间中几何对象的面积、长度、距离、周长、质心 |

# 1.5 现有空间数据库管理系统产品简介

---

- Oracle Spatial
  - <https://docs.oracle.com/database/121/SPATL/toc.htm>
- DB2 Spational Extender
  - [https://www.ibm.com/support/knowledgecenter/en/SSEP\\_GG\\_11.1.0/com.ibm.db2.luw.spatial.topics.doc/doc/csbp1001.html](https://www.ibm.com/support/knowledgecenter/en/SSEP_GG_11.1.0/com.ibm.db2.luw.spatial.topics.doc/doc/csbp1001.html)
- SQL Server Spatial
  - <https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-sql-server>
- MySQL Spatial
  - <https://dev.mysql.com/doc/refman/5.7/en/spatial-type-overview.html>
- PostgreSQL PostGIS

# 1.5 现有空间数据库管理系统产品简介

- 国内数据库引擎开发团队

- 大型通用数据库系列：人大金仓、达梦、神州通用、南大通用
- 腾讯系：TDSQL、TXSQL、Tbase、PhxSQL
- 阿里系三个团队：阿里巴巴集团数据库事业部、阿里云、Oceanbase
- 其他互联网：京东云、百度、小米
- 华为系三个团队：2012 高斯、2012 分布式实验室、华为云（IT 企业产品线）
- DB2 中国研发团队（曾经的存在）、EsgynDB 中国团队、国家电网、中国移动苏州研究院、中国电信广州团队（尚存在否？）
- NewSQL 系列：PingCAP、巨杉
- PostgreSQL 系：亚信南京 AntDB、中兴 GoldenDB、Greenplum 中国团队、飞象
- MySQL 系列：爱可生、上海热璞、万里开源、MySQL 中国区研发成员、OneSQL
- 分析型系列 / 大数据系列：柏睿数据 RapidsDB、酷克数据、偶数科技、Kylin 创业团队 Kyligence、星环科技
- Informix 系列：华胜信泰、福建星瑞格、南大通用（重复）
- 其他：Haisql、Highgo db、许继集团 SG-RDB、Cedar、上容、天曦 TXDB、HHDB、博阳数据管理系统、东方国信、优炫云数据库、新华三、鼎天盛华 Huayisoft、HUABASE

# 1.5 现有空间数据库管理系统产品简介

| 名称            | 开发商       | 性质 | 类型    | 空间数据支持方式                     |
|---------------|-----------|----|-------|------------------------------|
| Access        | Microsoft | 商业 | 关系型   | —                            |
| BerkeleyDB    | Oracle    | 开源 | 嵌入式   | —                            |
| DB2           | IBM       | 商业 | 关系型   | 数据库底层+ArcSDE中间件              |
| Informix      | IBM       | 商业 | 关系型   | 数据库底层+ArcSDE中间件              |
| Ingres        | Ingres    | 开源 | 关系型   | 数据库底层                        |
| Interbase     | Borland   | 开源 | 关系型   | —                            |
| MaxDB         | MySQL     | 开源 | 关系型   | —                            |
| MonetDB       | 学术机构      | 开源 | 列式    | —                            |
| MySQL         | MySQL     | 开源 | 关系型   | MySQL Spatial                |
| Oracle        | Oracle    | 商业 | 关系型   | 数据库底层+ArcSDE中间件              |
| PostgreSQL    | P Global  | 开源 | 关系型   | PostGIS扩展                    |
| SQL Anywhere  | Sybase    | 商业 | 移动嵌入式 | 用于移动设备                       |
| SQL Server    | Microsoft | 商业 | 关系型   | 数据库底层+ArcSDE中间件              |
| SQLite        | Richard   | 开源 | 嵌入式   | —                            |
| Sybase        | Sybase    | 商业 | 关系型   | —                            |
| Teradata      | Teradata  | 商业 | 关系型   | Teradata geospatial solution |
| Versant       | Versant   | 商业 | 面向对象型 | —                            |
| Visual FoxPro | Microsoft | 商业 | 关系型   | —                            |

# 1.5 现有空间数据库管理系统产品简介

- 国内自主开发数据库

| 名称       | 开发商  | 类型   | 是否支持空间数据 | 空间数据支持方式    |
|----------|------|------|----------|-------------|
| 金仓数据库    | 人大金仓 | 关系型  | 是        | SuperMap中间件 |
| 达梦数据库    | 达梦公司 | 关系型  | 否        | —           |
| OpenBASE | 东软集团 | 关系型  | 否        | —           |
| 神通数据库    | 神州通用 | 关系型  | 否        | —           |
| iBase    | 国信贝斯 | 非结构化 | 否        | —           |
| Gbase    | 南大通用 | 列式   | 否        | —           |
| HUABASE  | 华鼎数据 | 列式   | 否        | —           |
| BeyonDB  | 博阳世通 | 关系型  | 是        | 原生支持        |

# 第一章 地理空间数据库概论

---

- 1.1 关系数据库基本概念
- 1.2 空间数据库基本概念
- 1.3 空间数据管理技术的产生与发展
- 1.4 现有空间数据库标准简介
- 1.5 现有空间数据库管理系统产品简介 (自学)
  
- 空间计算思想
  - 空间关联
  - 时空查询
- 名词缩写:  
DBMS, RDBMS, SDBMS, ORDBMS,  
DDL, DML, SDDL, SDML,  
SQL, SSQl, SDB, DBA