



实验报告内容要求 – 大实验（10分）

1. 题目：包含算法名称
2. 摘要：两百字以内
3. 算法来源
4. 算法简介：流程图+文字
5. 算法实现：介绍编程模块，或者图示
6. 算法性能分析
 1. 文字介绍算法的优缺点
 2. 针对上述优缺点，准备设计怎样的实验测试？
7. 实验测试
 1. 测试图片、测试结果、原因分析
 2. 重点测试优点，缺点测试一两个即可
8. 小结：通过实验测试，给出自己对该算法的认识

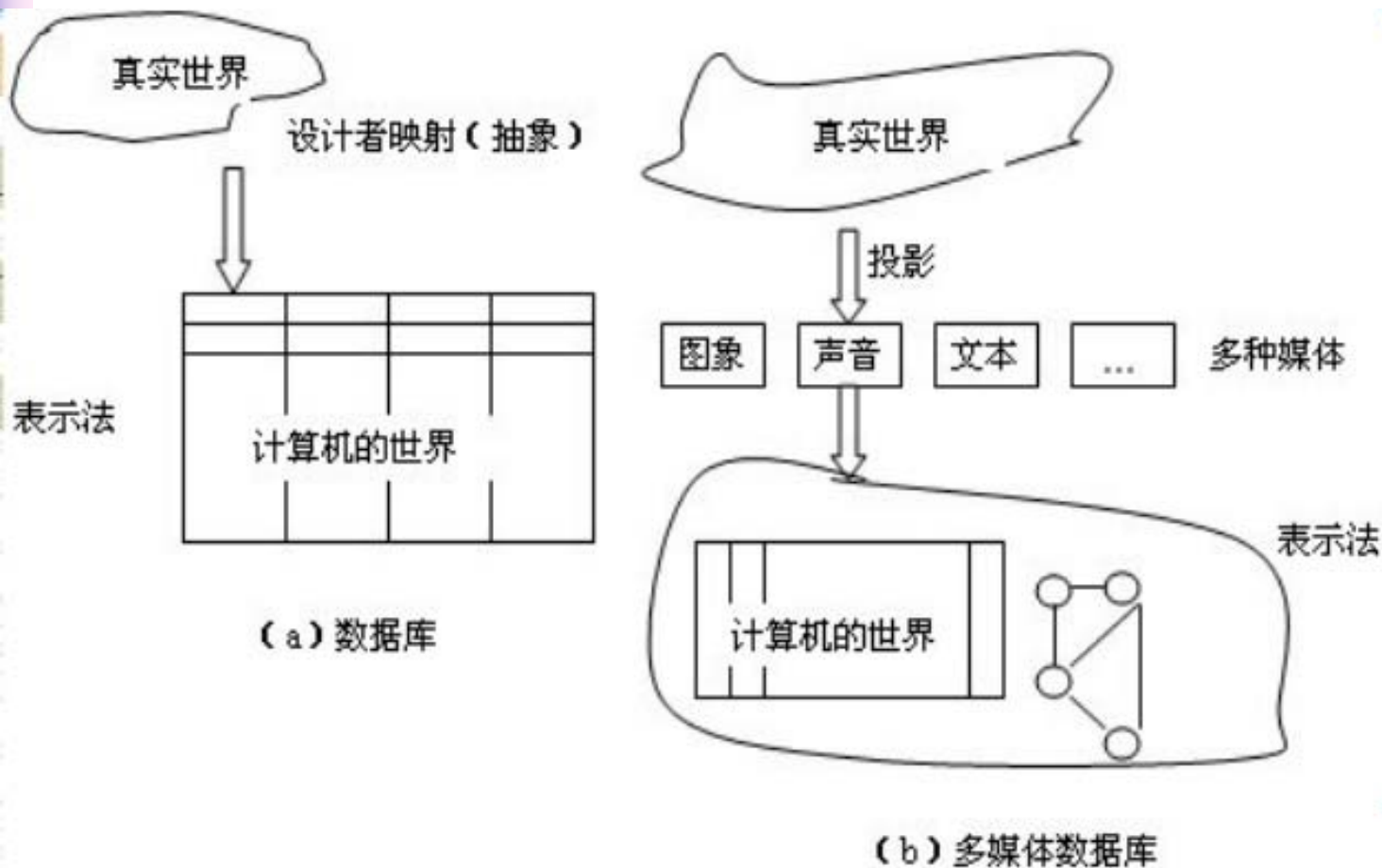


关于多媒体数据库的小结 – 与RDB对比分析

- 引入 & 内容管理需求
- 多媒体数据库技术的发展
- 数据模型&体系结构
- 多媒体查询语言 SQL/MM
- 多媒体数据库设计 及 研究问题

关于多媒体数据库

- **MMDB**: 最早由**Tsichritzis**和**Christodoulakis**于**1983**年在第九届**VLDB**会议上提出。





多媒体数据库 – 引入 & 内容管理需求

- **MMDBMS**

- 对非结构化媒体信息的管理
- 不同类型媒体、不同应用需求、基于内容的检索和查询

- **多媒体内容管理需求**

- 多媒体内容所具有的时空关系
 - 数据模型？索引结构？
- 对语义内容的识别和解释
 - 多媒体内容的描述和定义？
- 查询表现
 - 用户界面？人机交互？



多媒体数据库 – 多媒体数据库技术发展

- 20世纪90年代中期
 - 依赖操作系统进行存储和文件检索
- 20世纪90年代后期开始
 - 出现了除了多媒体内容的商业系统，对不同类型的媒体定义了复杂的对象类型。（**面向对象**方式 – 数据类型及操作）
 - **ORDBMS** – 对象关系数据库
 - **Oracle10g、IBM DB2、Informix、PostgreSQL、等**
- 正在进展的项目 – 强调语义（**MPEG7、MPEG21**）
 - **MARS**（**multimedia analysis and retrieval system**）



多媒体数据类型及编码

- 多媒体数据类型
 - 文本、图像、图形、声音/音频、视频、动画、3D、VR/AR/MR、元数据、.....
- 压缩编码技术
 - 必要性和有效性
 - 压缩标准
 - 对应的文件后缀名
 - 如图像对应的: jpg, bmp, gif, png, tiff,



多媒体数据库 – 数据模型

- 数据模型：数据的静态/动态特征、完整性约束
- 多级数据模型：概念、逻辑、物理
- **MMDB中的数据模型**
 - 具有灵活性，可对多媒体数据进行不同级别的抽象和计算
 - 可扩展
 - 能表示具有复杂时空关系的媒体数据及复合对象
 - 能进行有效存储和检索
- 目前没有通用的类型
 - 对不同数据类型定义不同结构及操作
 - 如，可变长字符、**BLOB**、**CLOB**、**IMAGE**、**TEXT**、.....
 - 如，会议纪要、信函、书、科技论文等



多媒体数据库 - 数据模型（续）

第一范式 不允许嵌套

- 扩充的关系模型
 - **NF²**数据模型 – **Non First Normal Form**
 - 表中可嵌套表；**BLOB**
- 面向对象数据模型
 - 引入面向对象概念，描述复杂对象
- 对象-关系数据模型
 - 用户定义新的数据类型和操作，并结合关系数据库提供的存储管理功能
- 面向具体应用 – 如，超文本数据模型
 - 定义文本的概念结构，实现高效管理



多媒体数据库 — 体系结构

- 联邦型
 - 多个独立媒体数据库整合在一起，各自具有独立的**DBMS**，
- 集中统一型
 - 多种媒体被统一存储在一个库内，由**MMDBMS**统一管理。
- 客户/服务器型
 - 多用于网络环境下。多种媒体数据库相对独立，通过专用服务器和一个多媒体管理服务器相连。
- 超媒体型
 - 多种媒体数据库分散存储在有网络连接的不同存储空间。重点在于对数据时空索引的组织。
- 云~~~

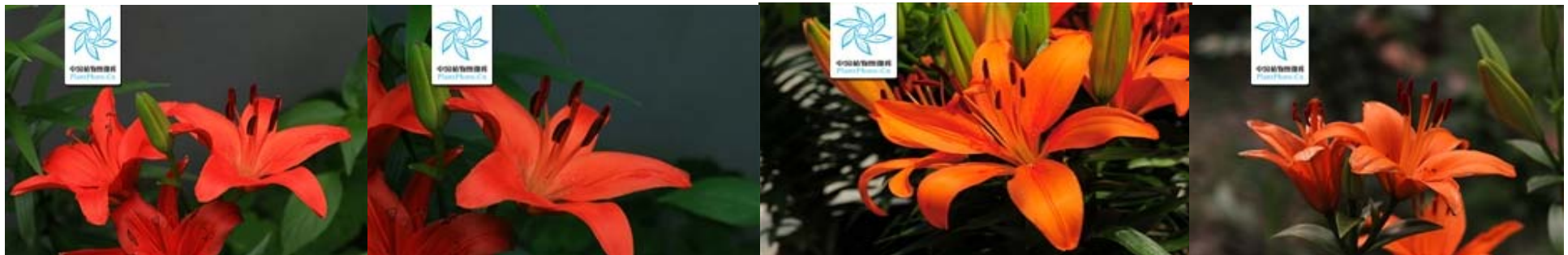


多媒体数据库 – SQL/MM

- 1992, ISO/IEC针对“文本、时空、静态图片和数据挖掘”的国际标准
 - 尽可能统一各研究机构/厂商在多媒体数据处理中的关键词
 - 定义框架、定义几种SQL对象类的“类库”
 - SQL/MM由六部分构成
 - 第一部分：框架 **Framework**
 - 第二部分：全文 **Full Text**
 - 第三部分：空间 **Spatial**
 - 第五部分：静态图像 **Still Image**
 - 第六部分：数据挖掘 **Data Mining**
 - 部分数据库系统提供支持

多媒体数据库 – SQL/MM - StillImage

- Oracle10g的SI_StillImage类型数据
 - *myimage SI_StillImage;*
 - *myAvgColor:=si_findavgclr(myimage);--颜色平均*
 - *myColorHist:=si_findclrhstgr(myimage);--颜色直方图*
 - *myPosColor:=si_findpstnlclr(myimage);--颜色位置*
 - *myTexture:=si_findtexture(myimage);--纹理*
 - 求解
 - *myFeatureList := new SI_FeatureList(myAvgColor, 0.25, myColorHist, 0.35, myPosColor, 0.10, myTexture, 0.5);*





多媒体数据库 – 多媒体数据库设计

- 体系结构 – 灵活和可扩展
- 界面设计 – 可用性
- 信息空间表示 – 逻辑信息空间和物理信息空间
- 可视推理 – 基于视觉表现线索的推测和制作推理的过程
- 相关反馈技术 – 与信息系统的交互
- 高维特征索引技术研究
-



多媒体数据库所面临的问题

- 数据的组织和存储
 - 数据量大、数据本身冗余大
- 媒体种类的增加
- 查询问题 — 信息检索
- 用户接口
- 信息的分布对体系结构的影响
- 长事务增多
- 服务质量
- 版本控制、等



研究问题 - 用户查询界面

- 用户对内容的感知
 - 用户与系统的交互方式
 - 查询界面
-
- 现代多媒体信息系统的一个重要特征就是信息获取过程的可交互性，人在系统中是主动的。



研究问题 - 高维索引技术

- * 高维特征向量的多维索引结构
 - * 如 **k-d**树和**R**-树，及改进的索引树结构
 - * 仍需有效的高维索引方法，以支持多特征、异构特征、权重、主键特征方面的查询要求。
- * 相似度匹配
 - * 聚类和神经网络方法适于解决这类问题。
- * 时间序列媒体
 - * 考虑它们的时间索引结构



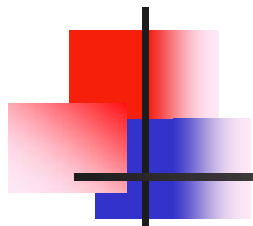
研究问题 – 高层概念和底层特征的关联

- 高层：人们倾向于使用概念表达事物。
- 底层：目前特征提取和检索都是基于底层特征方面的研究。
- 特定应用领域可实现：例如人脸识别和指纹识别，做到了一定程度的底层的特征与高层语义概念的关联
- 对于一般性的特征，建立起这种关联是非常困难的。



研究问题 - 多媒体信息安全

- 数字媒体内容的安全问题是制约多媒体内容市场的瓶颈问题，包括安全传递、访问控制和版权保护
- 数字版权管理技术：数字水印



多媒体数据库 期末总复习

2020/5/24



数据库系统概述 – 术语

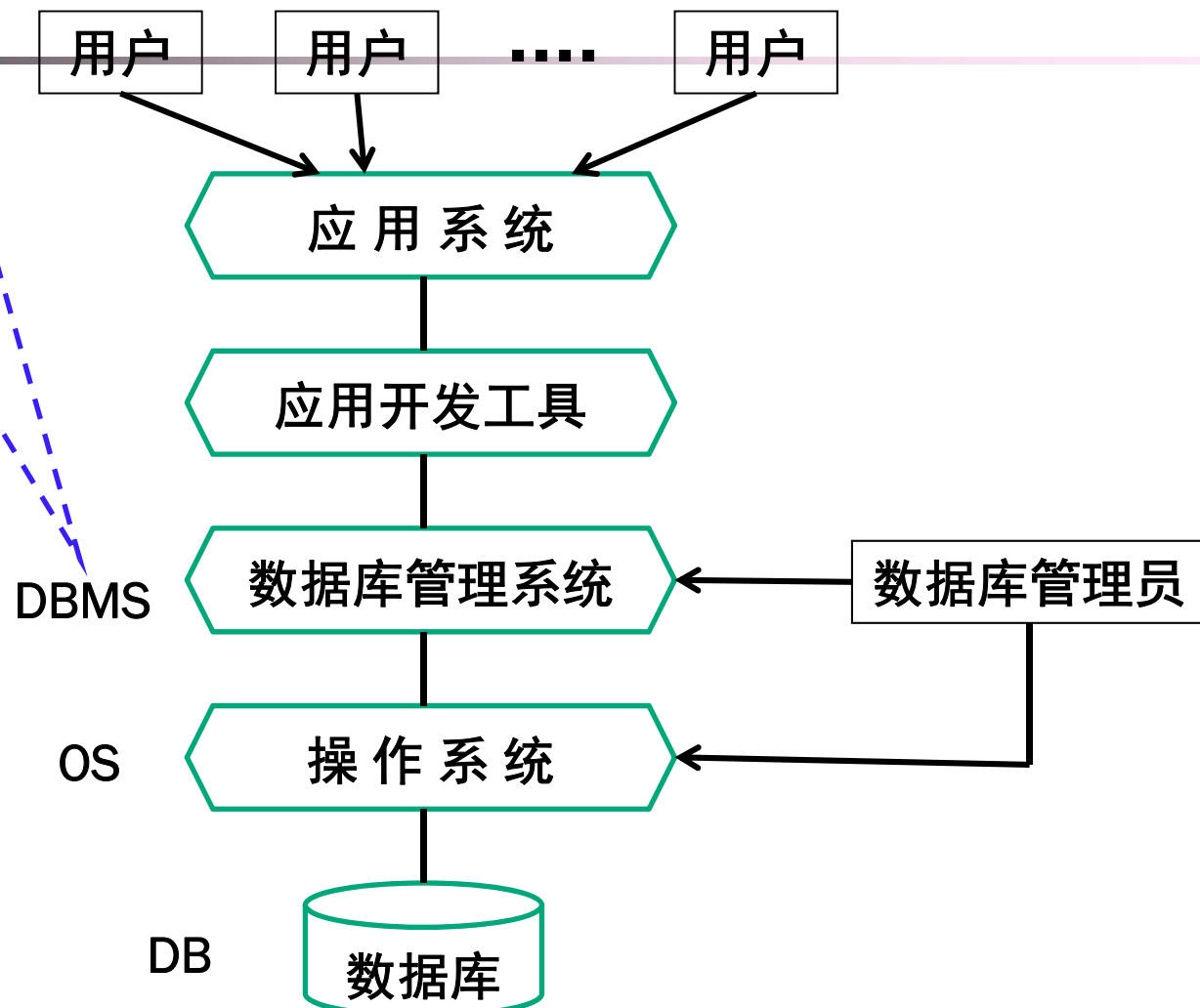
■ 常用术语

- 数据 **Data**
- 数据库 **DataBase**, 简称**DB**
- 多媒体数据库 **Multimedia Database**, 简称**MMDB** 计算机: 架构...
- 数据库管理系统 **DataBase Management System**, 简称**DBMS** 范指
- 数据库系统 **DataBase System**, 简称**DBS**
- 数据库管理员 **DataBase Administrator**, 简称**DBA**
- 元数据 **MetaData**, “数据的数据”

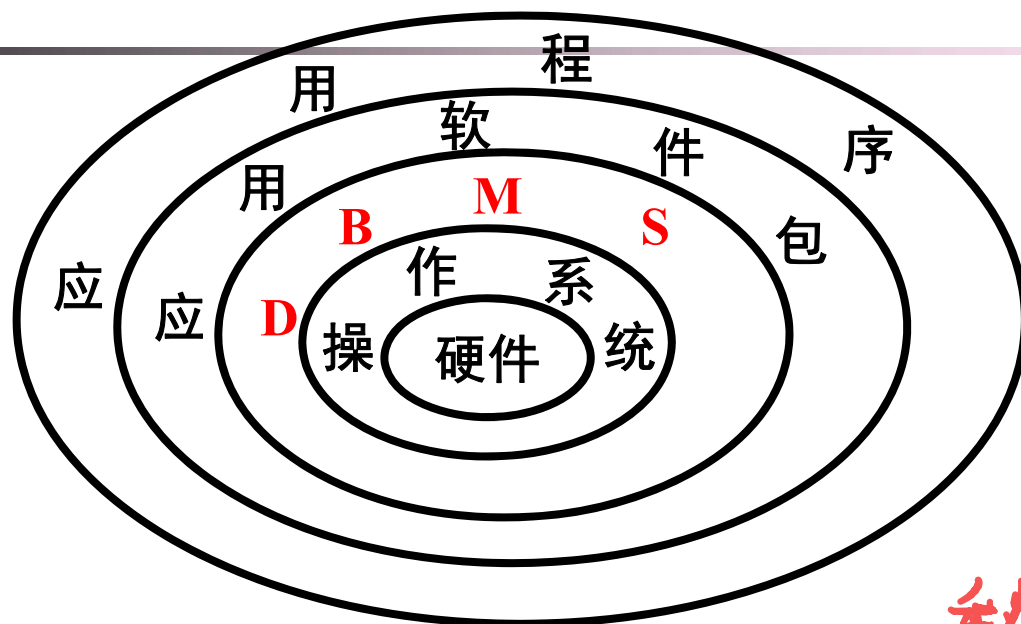
数据库系统 - DBS

DBMS主要功能

1. 数据定义功能
2. 数据组织、存储和管理
3. 数据操纵功能
4. 数据库事务管理和运行管理
5. 数据库的建立与维护
6. 其他功能



数据库在计算机系统的位置



秘书

DBMS: 用户与数据的接口。介于用户与操作系统之间的一层数据管理软件。为用户或应用程序提供访问DB的方法，包括DB的建立、查询、更新及各种数据控制。



数据库技术的发展概况

- **数据库技术**是应数据管理任务的需求而产生的。
- 数据管理技术的四个阶段：
 - 人工管理阶段（20世纪50年代中期以前）
 - 文件系统阶段（20世纪50年代后期～60年代中期）
 - 数据库系统阶段（20世纪60年代后期～今）
 - 高级数据库技术阶段（20世纪80年代中后期～今），也称第三代数据库技术。



数据库系统的特点

- 数据结构化：以关系数据库为例
 - 学生纪录：学号、姓名、性别、年龄、.....
 - 课程记录：课程编号、名称、学时、教材、.....
 - 学生选课：学号、课程编号、学期、成绩、.....
- 数据的共享程度高、冗余度低、易扩充
- 数据独立性高
- 数据由**DBMS**统一管理和控制
 - 数据的安全性保护、完整性约束、并发控制、数据库恢复、等



多媒体数据库的引入

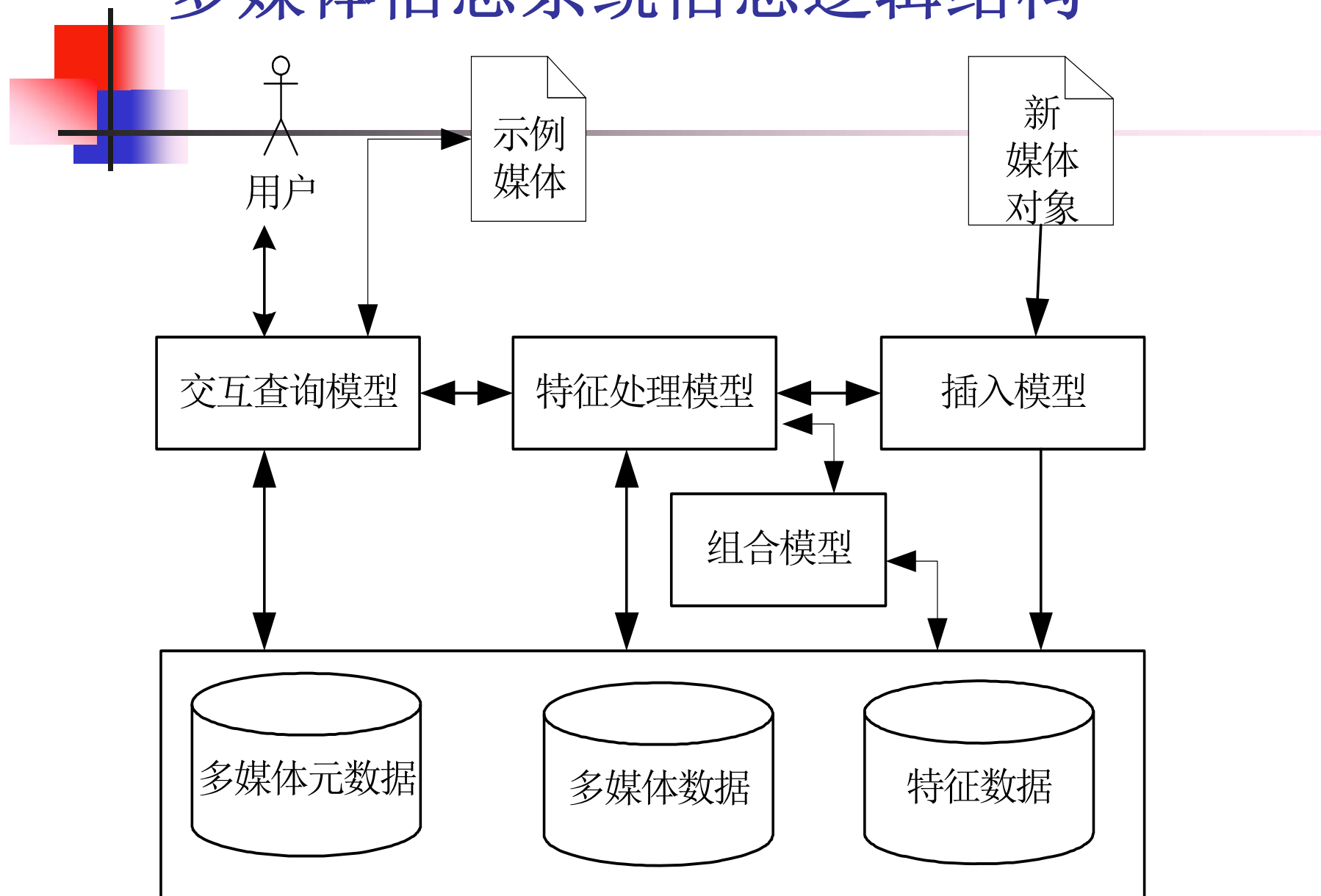
- 多媒体数据库 – **Multimedia Database**
- 多媒体数据特点
 - 数据量大、彼此关联 – 难以组织和存储
 - 类型多样 – 难以及时扩展
 - 含义模糊且具有较强的主观性 – 难以定义
 -
- 传统数据库 – 结构化数据管理
 - 缺乏管理时空关系的能力
 - 缺乏解释原始数据语义内容的能力
 - 查询表现能力
- 多媒体数据库 – **基于内容**的检索与查询



信息检索技术 - IR

- **Information Retrieval**: 是根据用户的查询, 检索对用户有用或相关的信息内容, 强调的是信息检索, 而不是数据检索。
- 与DBMS不同之处
 - **DBMS**: 结构化数据; 精确匹配检索数据项
 - **IR**: “相关性-relevance”是其核心问题
- 主要处理信息内容的表现、存储、组织和访问问题。
 - “检索关于意甲球队信息的网页, 要求: (1) 拥有巴西外援; (2) 具有参加欧洲联赛资格。 (3) 将查询结果按最近三年的意甲排名顺序的进行相关性排序。”
- IR检索技术还不够, 还需要多媒体信息检索技术
 - **Multimedia Information Retrieval System**

多媒体信息系统信息逻辑结构





数据模型 — 基本概念

- 模型：对现实世界中某对象特征的模拟和抽象。如，航模
- 数据模型：对现实世界数据特征的抽象，描述/组织/操作数据。精确描述了系统的静态特征、动态特征和完整约束性条件。
 - **DM: Data Model**
- **DM应满足的三个要求**
 - 比较真实地模拟现实世界
 - 容易为人所理解
 - 便于在计算机上实现
 - 故，针对不同的应用对象和目标，采用不同的数据模型

数据模型 — 两类不同的数据模型

■ 不同阶段采用不同DM

概念模型：也称信息模型，按照用户观点来对数据和信息建模，用于数据库设计

用户

现实世界

概念化

信息世界

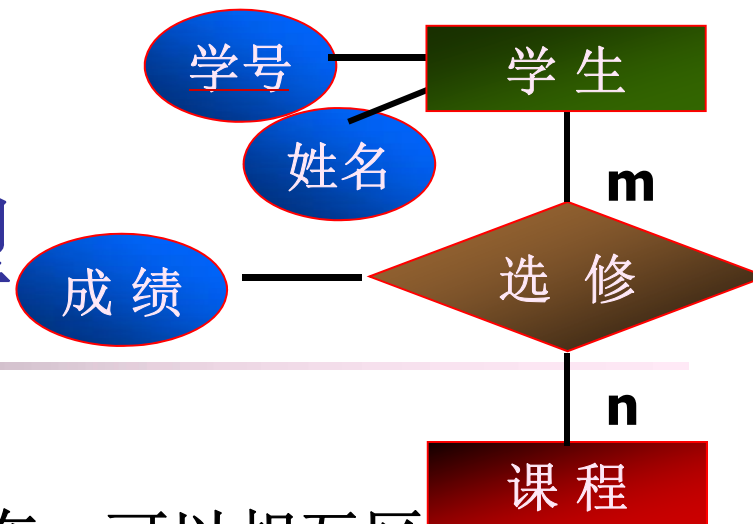
形式化

逻辑模型：照计算机观点来对数据建模，用于**DBMS**实现；
物理模型：面向计算机系统，实现数据在系统内部的具体存储

计算机

计算机世界

数据模型 — 概念模型



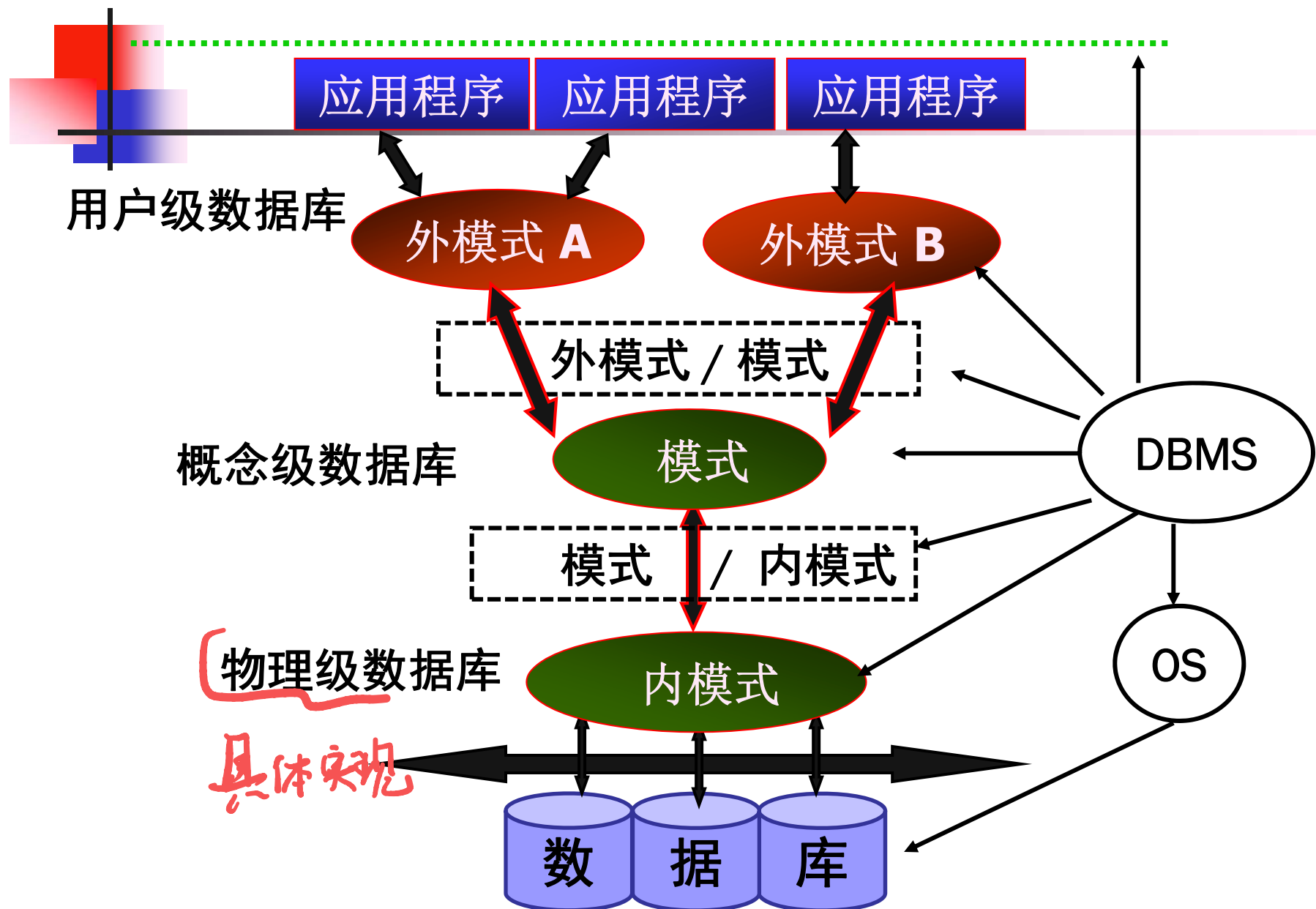
- 信息世界中的基本概念
 - 实体 (**E**ntity)：客观存在，可以相互区别的东西称为实体。实体可以是具体的对象，也可以是抽象的事件。
 - 属性 (**A**tttribute)：实体的某一方面的特征
 - 码(**K**ey)：唯一标识每个实体的属性或属性集
 - 联系 (**R**elationship)：实体内部、实体间的联系
- 实体联系模型 (**E**ntity **R**elationship **M**odel)。该模型直接从现实世界中抽象出实体类型及实体间联系，然后用**E-R**图表示的数据模型。

数据模型

学号	姓名	年龄	性别	系名	年级
2005004	张三	19	男	社会学	2005
2005006	李四	20	男	法律	2005
2005010	王五	18	女	计算机	2005
.....

- 1970年IBM公司研究院E.F.Codd首次提出数据库系统的关系模型。1981年ACM图灵奖
- 基本术语
 - 关系(Relation): 一张（二维）表格
 - 元组(Tuple): 表中一行
 - 属性(Attribute): 表中一列
 - 码(Key): 唯一确定一个元组的某个属性组
 - 域(Domain): 属性的取值范围
 - 分量: 元组中的一个属性值
 - 关系模式: 对关系的描述 – 表头
 - 如, 学生(学号, 姓名, 年龄, 性别, 系名, 年级)

数据库系统结构 — 三级模式、两级映像





数据库系统结构 — 三级模式

- 模式**Schema**：数据逻辑结构和特征的描绘，与具体取值相比，是相对稳定的。
- 三级模式
 - 模式：逻辑模式。库中**全部**数据的逻辑结构和特征的描述，是所有用户的公共数据视图。
 - 即不涉及数据的物理存储细节、也与具体应用程序等无关
 - 外模式：用户模式。用户能看到的、与某个应用程序有关的**局部**数据的逻辑结构和特征描述。
 - 保护数据库安全的有力措施
 - 内模式：存储模式。是数据在数据库**内部的表示**，如存储结构、索引方式、是否加密等。



数据库系统结构 — 两级映像

- 保证数据库系统中的数据具有较高的逻辑独立性和物理独立性。
 - 外模式/模式映像 — 逻辑独立性
 - 对于每个外模式，由该映像定义这个模式之间的对应关系，以确保模式改变时，该映像做相应改变即可，不需修改外模式，也就不需修改应用程序。
 - 模式/内模式映像 — 物理独立性
 - 模式/内模式映像是唯一的，定义了数据全局逻辑结构与存储结构之间的对应关系。当数据库存储结构改变时，可通过该映像的相应改变来确保模式不需改变，从而应用程序也就不需改变了。



关系数据库 — 数据结构 — 基本概念

- **关系(Relation)**
 - $D_1 \times D_2 \times \dots \times D_n$ 的有意义的子集称为在域 D_1, D_2, \dots, D_n 上的关系，记为 $R(D_1, D_2, \dots, D_n)$ 。
 - R 为关系的名； n 为关系的度（目）； $r \in R$ 表示 r 是 R 中的元组
- **候选码(Candidate key)**
 - 值能唯一标识一个元组的属性组，且不含多余属性，称该属性组为候选码。
- **主码(Primary key)**
 - 一个关系有多个候选码时，选定其中的一个作为主码。
- **外部码(Foreign key)**
 - 关系 R 的某一属性组 X 不是 R 的码，但是其他某一关系的码，称 X 为 R 的外部码。
- **主属性(Prime attribute)**
 - 关系 R 中构成候选码的属性称为主属性。



关系数据结构 – 关系数据库

- 在关系模型中，实体以及实体间的联系都是用**关系来表示**。在一个给定的现实世界领域中，相应于所有实体及实体之间的联系的关系的集合构成一个关系数据库。
- 关系数据库的“型”与“值”
 - “型”：关系数据库模式，对库的描述。是稳定的、静态的。
 - “值”：关系模式在某一时刻对应的关系的集合。是动态的，随时间而改变的。



关系操作 — 基本操作

- 关系操作特点：集合操作方式
 - 操作对象和结果都是集合，一次一集合（set-at-a-time）
- 基本操作
 - 查询Query、插入Insert、删除Delete、修改Update
- 查询操作：最主要的部分
 - 选择Select、投影Project、连接Join、除Divide、并Union、差Except、交Intersection、笛卡儿积、等 — 了解即可



SQL - Structured Query Language

1. **高度非过程化的语言**：用户只需提出“干什么”，至于“怎么干”由DBMS解决；用户只需要在查询语句中提出需要什么，DBMS即可按路径存取，并把结果返回给用户。
2. **面向集合的语言**：每一个SQL的操作对象是一个或多个关系，操作的结果也是一个关系。
3. **一种语法结构，两种使用方式**：即可独立使用，又可嵌入到宿主语言中使用，具有自主型和宿主型两种特点。
4. **具有查询、定义、操作和控制四种语言一体化的特点**。它只向用户提供一种语言，但该语言具有上述多种功能，可独立完成数据库生命周期中的全部活动。

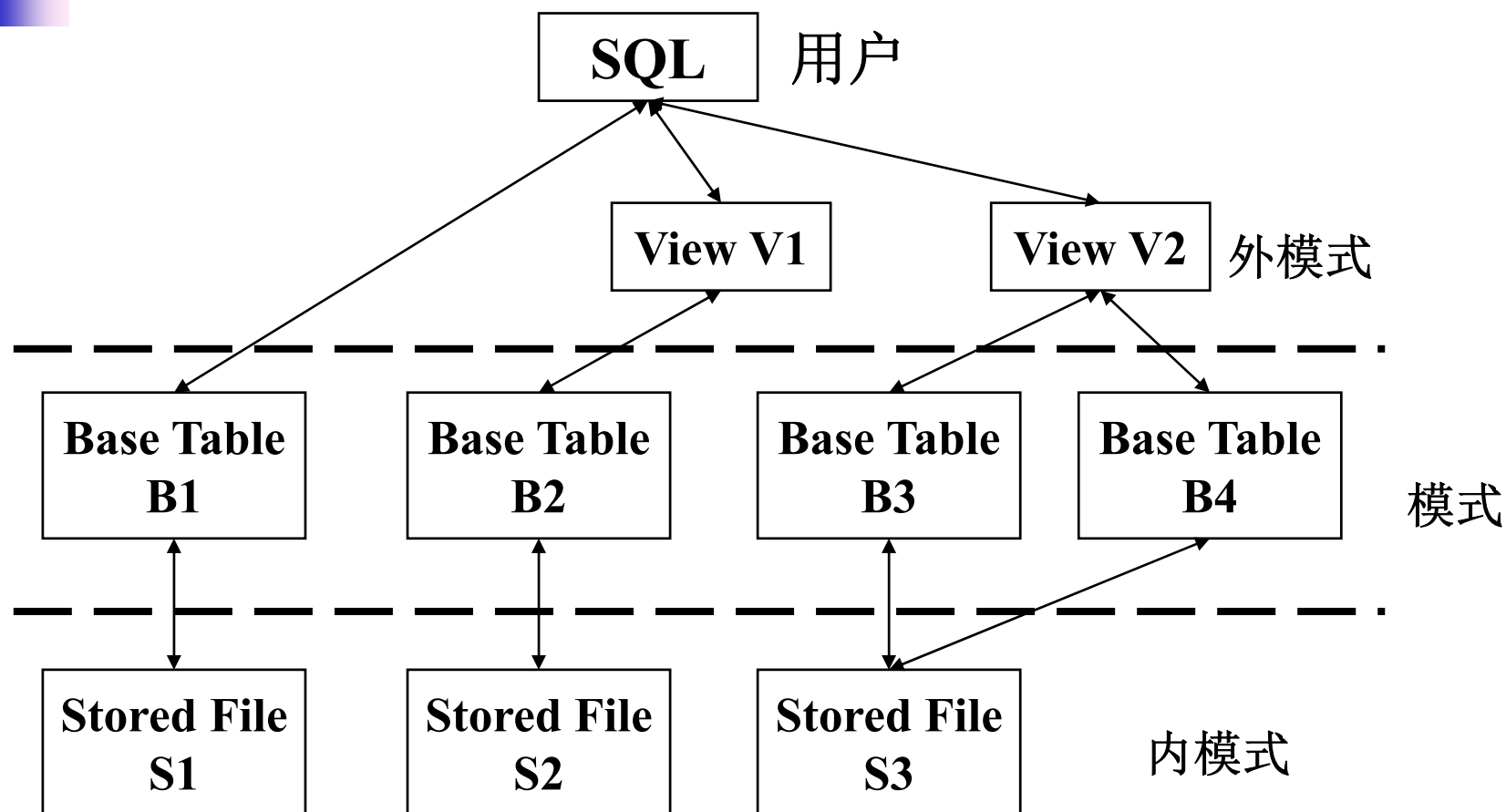


SQL的特点（续）

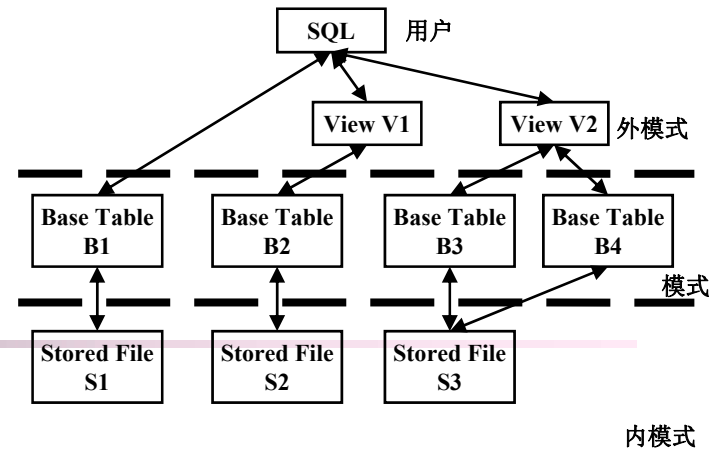
5. 语言简洁、易学易用：核心功能只有9个动词，语法简单，接近英语。

SQL功能	动词
数据查询	SELECT
数据定义	CREATE, DROP, ALTER
数据操纵	INSERT, UPDATE, DELETE
数据控制	GRANT, REVOKE

SQL语言支持的关系数据库的三级模式结构



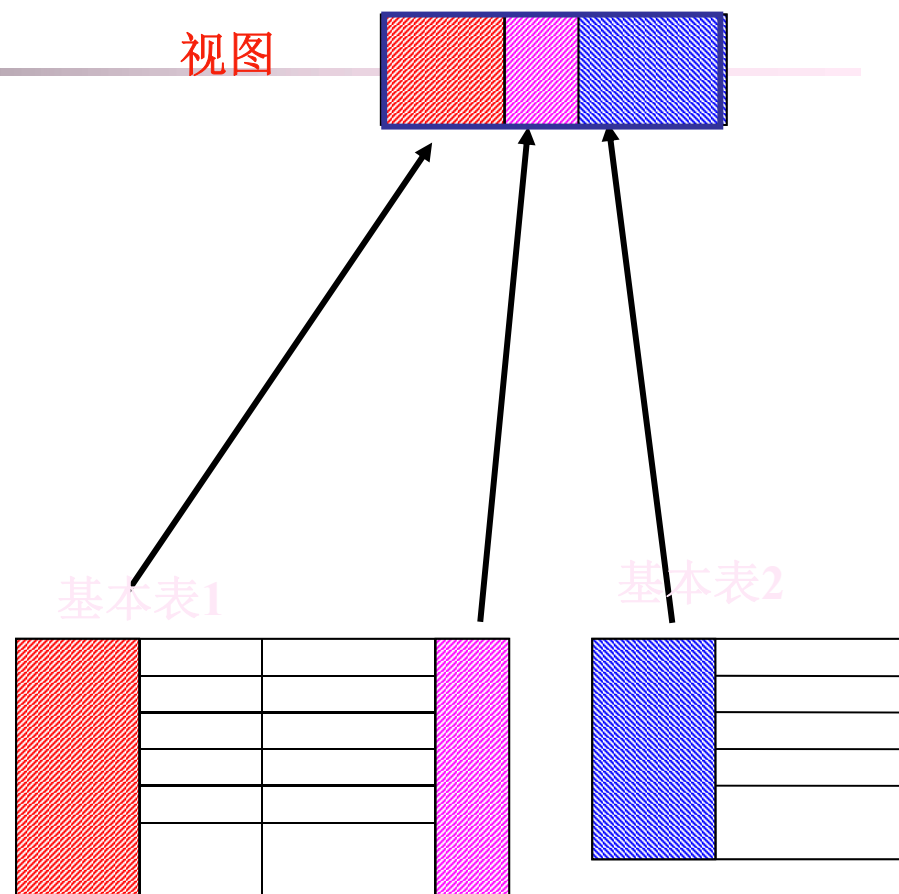
基本概念 - 视图、基本表



- 用户可以用SQL语言对视图(View)和基本表(Base Table)进行查询等操作，在用户观点里，视图和表一样，都是关系。
- **视图**是从一个或多个基本表中导出的表，本身不存储在数据库中，只有其定义，可以将其理解为一个虚表。
- **基本表**是本身独立存在的表，一个（或多个）基本表对应一个存储文件，一个表可以带若干索引，存储文件及索引组成了关系数据库的内模式。

视图概念

- 视图是一个虚表
- 数据库中只存放视图的定义
- 视图对应的数据仍存放在原来的表中
- 随着表中数据的变化，视图的数据随之改变。
- 对视图的查询与基本表一样
- 对视图的更新将受到一定的限制



视图概念示意图



视图的优点

- 提供数据的逻辑独立性
- 提供数据的安全保护功能
- 简化用户的操作
(对系统构成的视图，用户不必关心各表间的联系)
- 同一数据多种用法



数据查询SQL – 语句格式

■ **SELECT** [ALL|DISTINCT] [表名.]{*|列名|表达式[AS 新列名]}

[INTO :主变量1[,主变量2]...]

FROM {表名或视图名} ...

[WHERE 条件表达式|子查询]

[GROUP BY 列名1, ...[HAVING 分组表达式]]

[{UNION|INTERSECT |EXCEPT}SELECT...FROM...]

[ORDER BY 列名|列序号[ASC|DESC], ...];



数据查询SQL – 语句执行过程

SELECT...	⑤ 投影
FROM...	① TABLE→内存
WHERE...	② 选取元组
GROUP BY...	③ 分组
HAVING...	④ 选择分组
[{UNION ... }	⑥ 查询结果的集合运算
SELECT...]	①~⑤
ORDER BY.....	⑦ 排序输出



数据库安全性 - 引入及基本概念

- 数据库特点：数据共享
 - 1. 数据共享所引发的数据库安全问题
 - 2. 数据共享不能是无条件的共享
- 由**DBMS**统一提供数据库的数据保护功能
 - 保证数据库内数据的安全可靠和正确有效
 - **安全性**：防止因用户非法使用数据库造成数据泄露、更改或破坏。
 - **完整性**：防止出现不正确的数据（不符合语义的数据）
- 共享的庞大数据的安全问题尤为重要！
- 系统安全保护措施是否有效，是数据库系统的重要技术指标之一。



数据库安全性控制的常见策略

- 用户标识与鉴别
- 存取控制
 - 自主存取控制：**DAC**，面向用户
 - 数据库角色
 - 强制存取控制：**MAC**，用户和数据的严格匹配
- 视图机制
- 审计
- 数据加密



统计数据库 — 基本概念

- 概念：管理统计数据的数据库系统
 - 统计数据库的特点
 - 包含大量数据内容
 - 目的：向用户提供各种统计汇总信息，而不是单个记录信息。
 - 操作：允许用户查询聚集类型的信息（例如合计、平均值等），但不允许查询单个记录信息
- 例：允许查询“程序员的平均工资是多少？”
不允许查询“程序员张勇的工资？”



统计数据库 — 安全性问题

■ 统计数据库中特殊的安全性问题

- 可能存在**隐蔽的信息通道**，使得用户能从合法的查询中推导出不合法的信息。

例：下面两个查询都是合法的：

1. 本公司共有多少女高级程序员？
2. 本公司女高级程序员的工资总额是多少？

规则1：任何查询至少要涉及N(N足够大)个以上的记录。

规则2：任意两个查询的相交数据项不能超过M个

规则n：

目标：试图破坏安全的人所花费的代价 >> 得到的利益



数据库完整性 — 基本概念

■ 数据库的完整性

安全性：防范非法用户和非法操作、防止对库内数据的非法存取。

- 数据的正确性和相容性
- 防止不符合语义、不正确的数据进入数据库。

例：学生的年龄必须是整数，取值范围为[14， 29]；
学生的性别只能是男或女；学生的学号一定是唯一的；
学生所在的系必须是学校开设的系；等等...

- **完整性控制**是DBMS核心功能之一
 - 减轻应用程序员的负担
 - 为所有用户和所有应用提供一致的数据库完整性

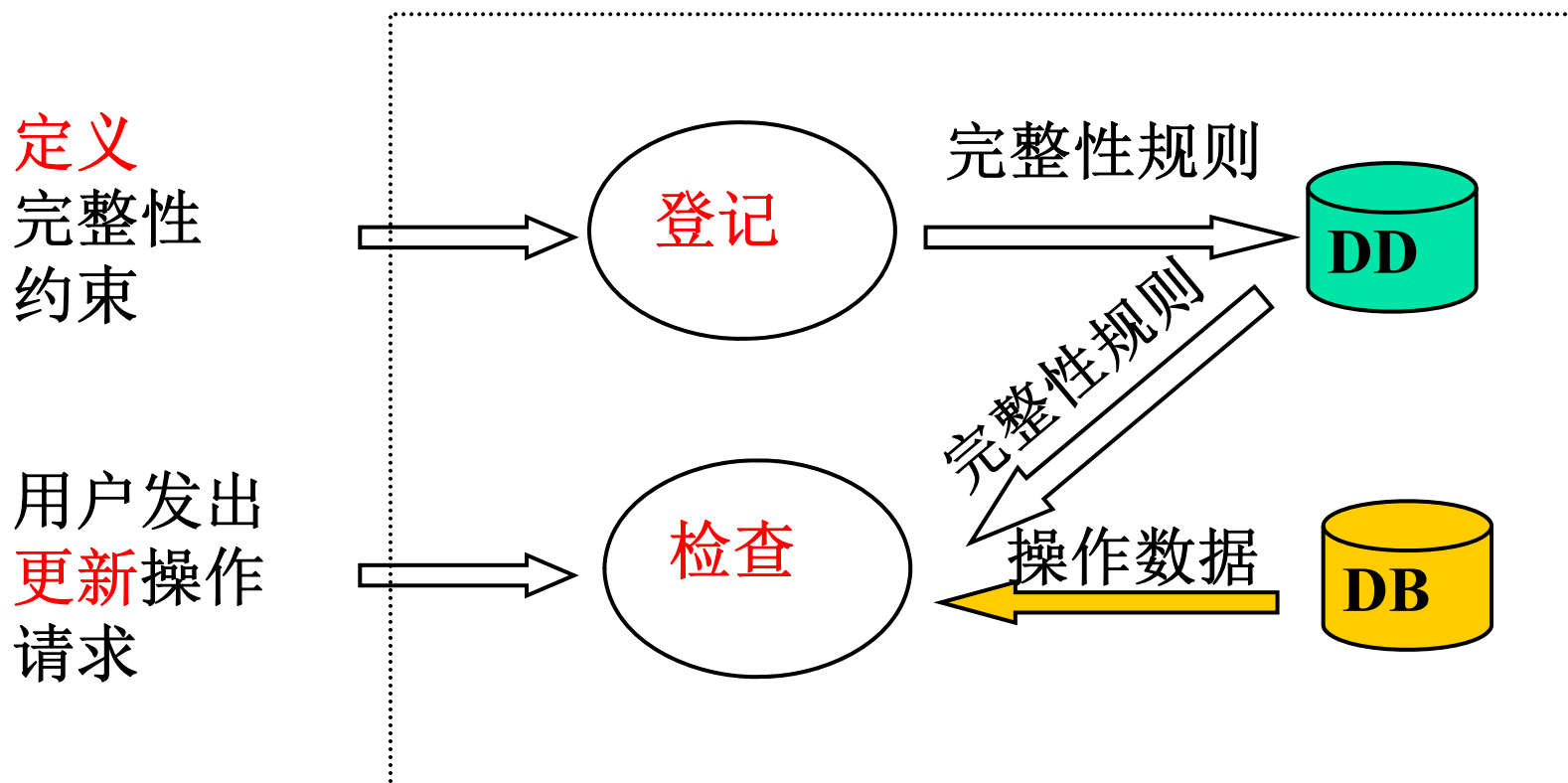


数据库完整性 — 基本概念

- **DBMS如何维护数据库的完整性**
 - 提供**定义**完整性约束条件的机制
 - **DDL**语句描述完整性（存入数据字典）
 - 提供完整性**检查**的方法
 - 检查**操作执行后**数据是否违背完整性约束条件
 - 检查时机：立即执行检查、延迟执行检查
 - 借贷平衡、多次操作完成一个**事务**
 - **违约处理**
 - 对违背完整性约束条件的操作，采取
 - 拒绝 **No Action**
 - 级联执行其他操作 **Cascade**
 -

完整性控制的实现原理 — 完整性控制子系统

- 定义、检查、采取行动、.....





主码和实体完整性

- 候选码：该属性组值能唯一标识关系中的一个元组。学号、姓名、性别、年龄、手机、...
- 主属性：候选码的诸属性
- 非主属性：不包含在任何候选码中的属性
- 主码/主键：从候选码中挑一个作为该关系的PRIMARY KEY
- 实体完整性：基本关系的主属性均不能为空



主码和实体完整性 — 定义

■ CREATE TABLE中的PRIMARY KEY

CREATE TABLE Student

(Sno CHAR(9) **PRIMARY KEY**,
Sname CHAR(20) NOT NULL,
Ssex CHAR(2),
Sage SMALLINT,
Sdept CHAR(20));

列级约束

CREATE TABLE Student

(Sno CHAR(9),
Sname CHAR(20) NOT NULL,
Ssex CHAR(2),
Sage SMALLINT,
Sdept CHAR(20)
PRIMARY KEY (Sno));

表级约束



外码和参照完整性

■ 外码

- 设F是基本关系R的一个/组属性，但不是关系R的码。 K_S 是基本关系S的主码。若F与 K_S 相对应，则称F是R的**FOREIGN KEY**。

■ 参照完整性规则

- 若属性（属性组）F是基本关系R的外码，它与基本关系S的主码 K_S 相对应，则对于R中每个元组在F上的值必须为
 - 或者取空值
 - 或者等于S中某个元组的主码值



外码和参照完整性 — 定义

■ CREATE TABLE中的FOREIGN KEY

CREATE TABLE SC

```
( Sno CHAR(9) NOT NULL,  
  Cno CHAR(4) NOT NULL,  
  Grade SMALLINT,  
  PRIMARY KEY (Sno, Cno),  
  FOREIGN KEY (Sno) REFERENCES Student(Sno),  
  FOREIGN KEY (Cno) REFERENCES Course(Cno)  
);
```



外码和参照完整性 — 违约处理

- 系统默认策略：拒绝执行。其他策略在创建表时显式说明

CREATE TABLE SC

(Sno CHAR(9) NOT NULL,

Cno CHAR(4) NOT NULL,

Grade SMALLINT,

PRIMARY KEY (Sno, Cno),

FOREIGN KEY (Sno) REFERENCES Student(Sno)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (Cno) REFERENCES Course(Cno)

ON DELETE NO ACTION /*不允许删除学生已选修的课*/

ON UPDATE CASCADE

);



面向具体应用 — 用户自定义完整性

- 用户定义的完整性 **User-defined Integrity**

- 用户自定义完整性是针对某一具体数据的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求，由应用环境决定。
- 如，要求某些非主属性也不能为空；学生成绩必须在0~100分之间，等等
- 完整性约束由数据库管理系统保证



过程性约束与触发器 — 概念

- 触发器（**Trigger**）：用户定义在关系表上的一类由事件驱动的特殊过程，一旦定义，任何触发该触发器的事件发生时，均由服务器自动激活相应的触发器。
- 特点
 - 类似约束，但比约束更加灵活
 - 可实现更为复杂的检查和操作
 - 具有更精细、更强大的数据控制能力



过程性约束与触发器 — 定义

■ 用CREATE TRIGGER命令建立触发器

CREATE TRIGGER <触发器名>

INSERT, UPDATE,
DELETE 或 组合

{ **BEFORE** | **AFTER** } <触发事件> **ON** <表名>

FOR EACH { **ROW** | **STATEMENT** }

[**WHEN** <触发条件>]

条件为真,方执行动作体

<触发动作体>

匿名PL/SQL过程块 或
已创建的存储过程



过程性约束与触发器 — 定义 — 例

- 定义一个**BEFORE**行级触发器，为教师表 **TEACHER** 定义完整性规则“教授的工资不能低于4000元，若低于，则自动改为4000元”

```
CREATE TRIGGER Insert_Or_Update_Sal
BEFORE INSERT OR UPDATE ON TEACHER
FOR EACH ROW
AS BEGIN
    IF (new.Job='教授') AND (new.Sal < 4000 ) THEN
        new.Sal := 4000;
    END IF;
END;
```



多媒体的不同类型

- 媒体特点、常用编码方法（文件后缀名）
 - 文本
 - 图形
 - 图像
 - 声音
 - 音频
 - 视频
 - 动画
 - 三维
 - **VR/AR/MR等**



文本检索中的术语权重计算: **tf.idf**加权法

- 术语频率 (Term frequency) :
 - term_i 在文档 d_j 中的出 次数, 记作 tf_{ij}
 - tf_{ij} 越高, term_i 于文档 d_j 越重要
 - 比如: 一篇乔丹的文章, 可以预期 “乔丹”、“飞人” tf 值会比较高
- 文档频率 (Document frequency) :
 - 含有 term_i 的文档的数量, 记作 df_i
 - df_i 越高, 意味着 term_i 在衡量文档之 相似性方面作用越低
 - 比如 “的” 的 df 肯定非常高, 但它不具有区别性, 称为 “非焦点词”



tf.idf加权法（续）

- 逆文档频率(Inverse document frequency: **idf.**)
 - 与 df_i 成“反比关系”，记作 idf_i

$$idf_i = \log\left(\frac{N}{df_i}\right)$$

- idf_i 越高，意味着 $term_i$ 于文档区别度越高
- N 全部文档的数量。如果一个术语仅出现在一个文档中， $idf_i = \log N$ ；如果一个术语出现在所有文档中， $idf_i = \log 1 = 0$



tf.idf加权法（续）

- 索引项加权：给那些经常出现在一个文档中，而不常出现在其他文档中的术语以更高的权重，即让“特别的词”从“一般的词”中凸现出来。
- 有许多不同的加权公式

公式一：

$$weight_{i,j} = tf_{i,j} \times idf_i = tf_{i,j} \times \log \frac{N}{df_i}$$

公式二：

$$weight_{i,j} = \begin{cases} (1 + \log(tf_{i,j})) \log \frac{N}{df_i}, & \text{当 } tf_{i,j} \geq 1 \\ 0, & \text{当 } tf_{i,j} = 0 \end{cases}$$

tf.idf加权法的应用实例

Query = “夏夜湖畔的蛙鸣”

D1 : 湖畔的夏夜常常很凉爽,

D2 : 湖畔有家“湖畔”啤酒花园, 花园中常常是鼓鼓的蛙鸣一片,

D3 : “蛙鸣”禅社举办“蛙鸣”诗会的消息.....

Term	...	湖畔	夏夜	的	常常	蛙鸣	禅社	诗会	...
<i>df</i>	...	2	1	3	2	2	1	1	...
<i>idf</i>	...	0.176	0.477	0	0.176	0.176	0.477	0.477	...

$$N = 3 \quad idf_i = \log\left(\frac{N}{df_i}\right)$$

文本处理中的索引结构 - 倒排索引

倒排索引 (inverted index) 是文档检索系统最常用的数据结构，用来存储在全文搜索中某个单词在一个文档或一组文档中存储位置的映射。带有倒排索引的文件称之为“**倒排索引文件**”

1 6 12 16 18 25 29 36 40 45 54 58 66 70

That house has a garden. The garden has many flowers. The flowers are beautiful

Vocabulary

Occurrences

beautiful

70

flowers

45, 58

garden

18, 29

house

6

内容 → 指定位置

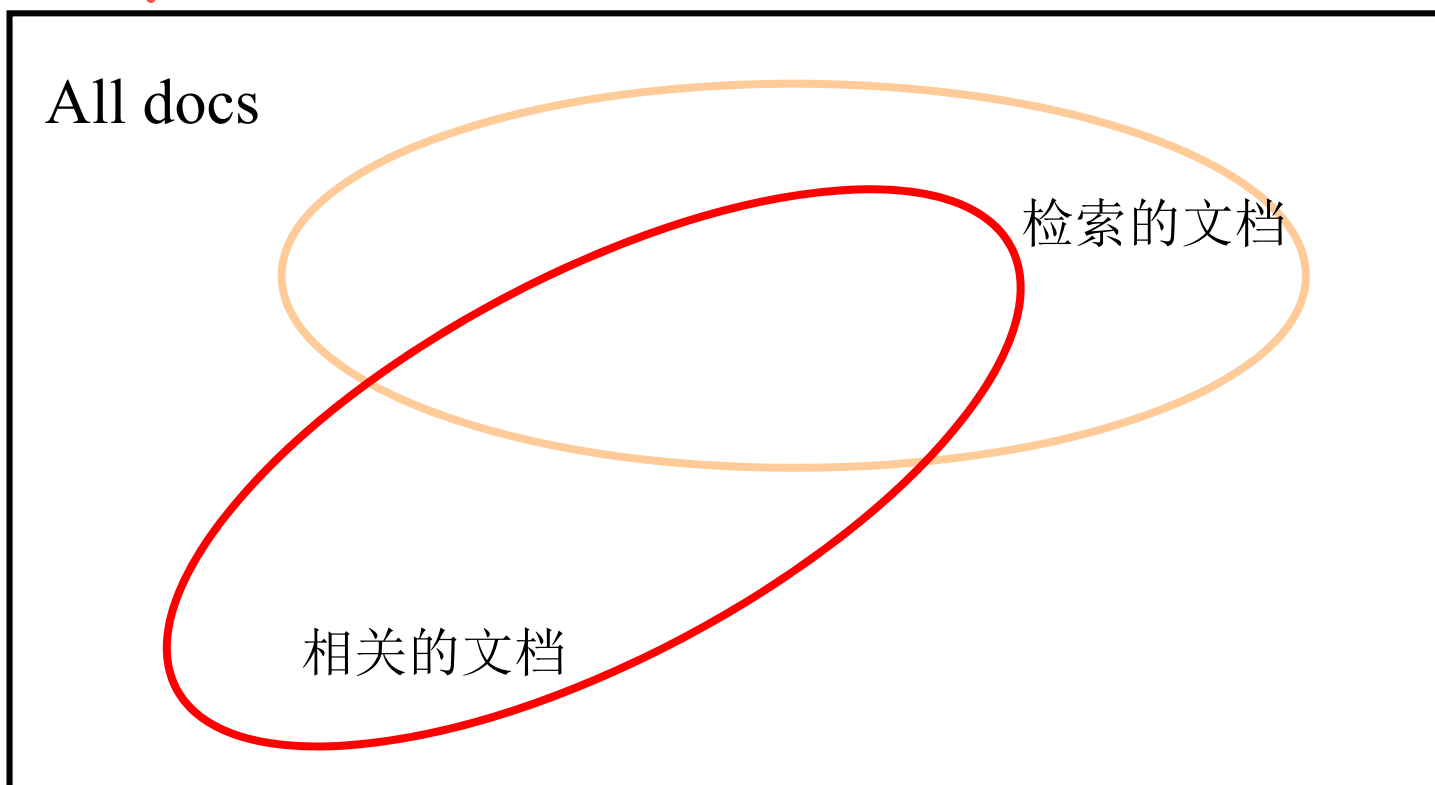
IR系统性能评估 — 准确率和召回率

$$\text{Precision} = \frac{|\text{RelRetrieved}|}{|\text{Retrieved}|}$$

准不准

$$\text{Recall} = \frac{|\text{RelRetrieved}|}{|\text{Rel in Collection}|}$$

全不全





评估指标

文档	属于检索结果集合 R_t	不属于检索结果集合 NR_t
NR 不相关	+ 不相关 + 在检索结果中 A	+ 不相关 + 不在检索结果中 C
R 相关	+ 相关 + 在检索结果中 B	+ 相关 + 不在检索结果中 D

$$\text{准确率} = \frac{\text{检索结果中和查询相关的文档数}}{\text{检索结果中的文档总数}} = \frac{B}{R_t} \quad \text{precision}$$

$$\text{召回率} = \frac{\text{检索结果中和查询相关的文档数}}{\text{文档库中所有和查询相关的文档数}} = \frac{B}{R} \quad \text{recall}$$

- 准确率度量查询的精确程度：准确率越高，不相关的文档越少
- 召回率度量查询的完整程度：召回率越高，漏检的文档越少



基于内容图像检索的引入

- 基于内容的图像检索的意义
 - “一幅图胜过千言万语”
 - 视觉信息是人们交流的重要形式
 - 图像是Internet上非常重要的一类媒体
 - 根据 **Online Computer Library Center (oclc.org)** 的统计，Web上有超过1亿的图像，并且以**50%~70%** 的年增长率增长。



基于内容检索 **VS** 基于文本/关键词检索

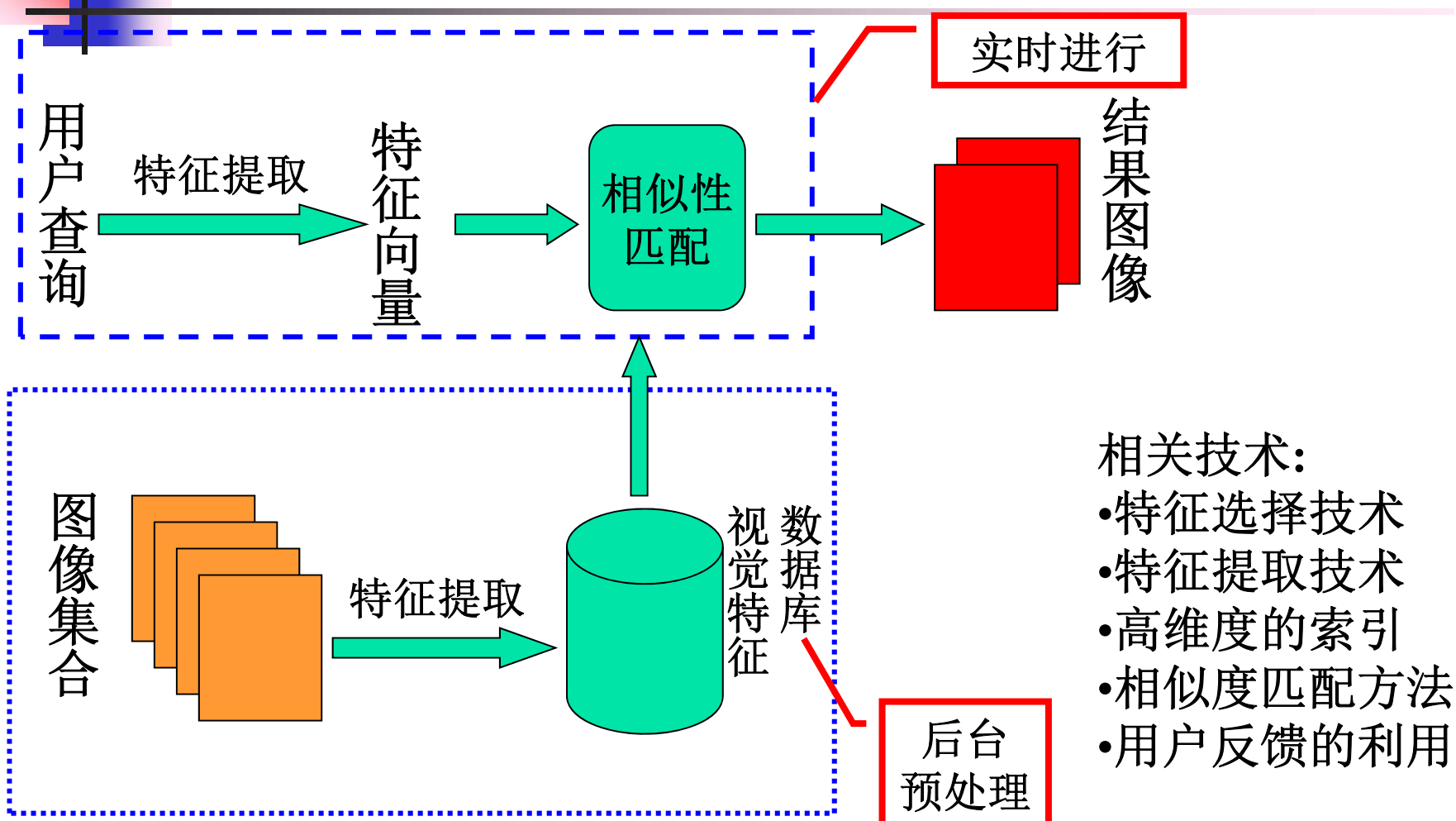
- 1979年，TBIR-基于文本的图像检索
- 1992年，形成共识“表示和索引图像信息的最有效方法应该是基于图像内容本身的”。
- 为什么不能通过文本标注来检索
 - 图像内容的自动注记仍然不可行
 - 手工注记工作量太大
 - 一幅图片可对应无数的描述
 - 一些视觉内容特征很难描述



基于内容的图像检索 - CBIR

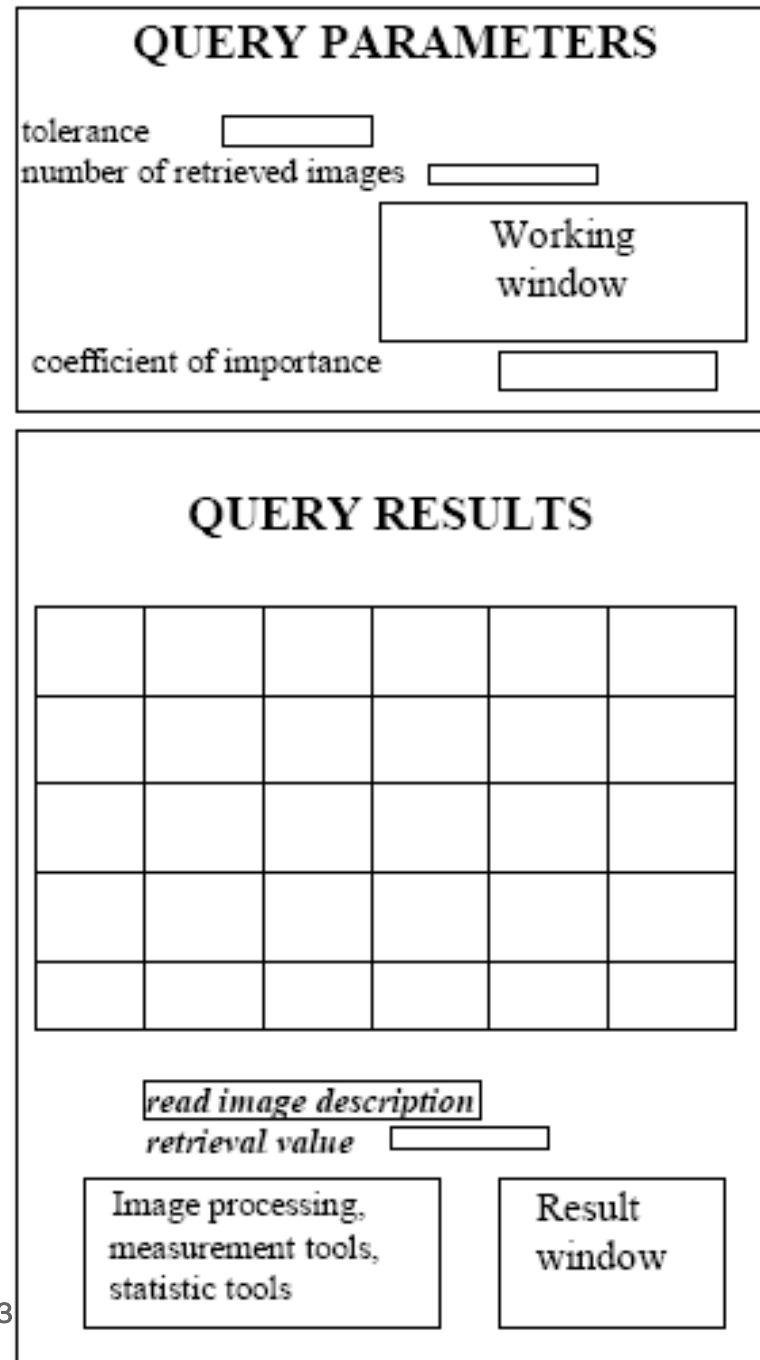
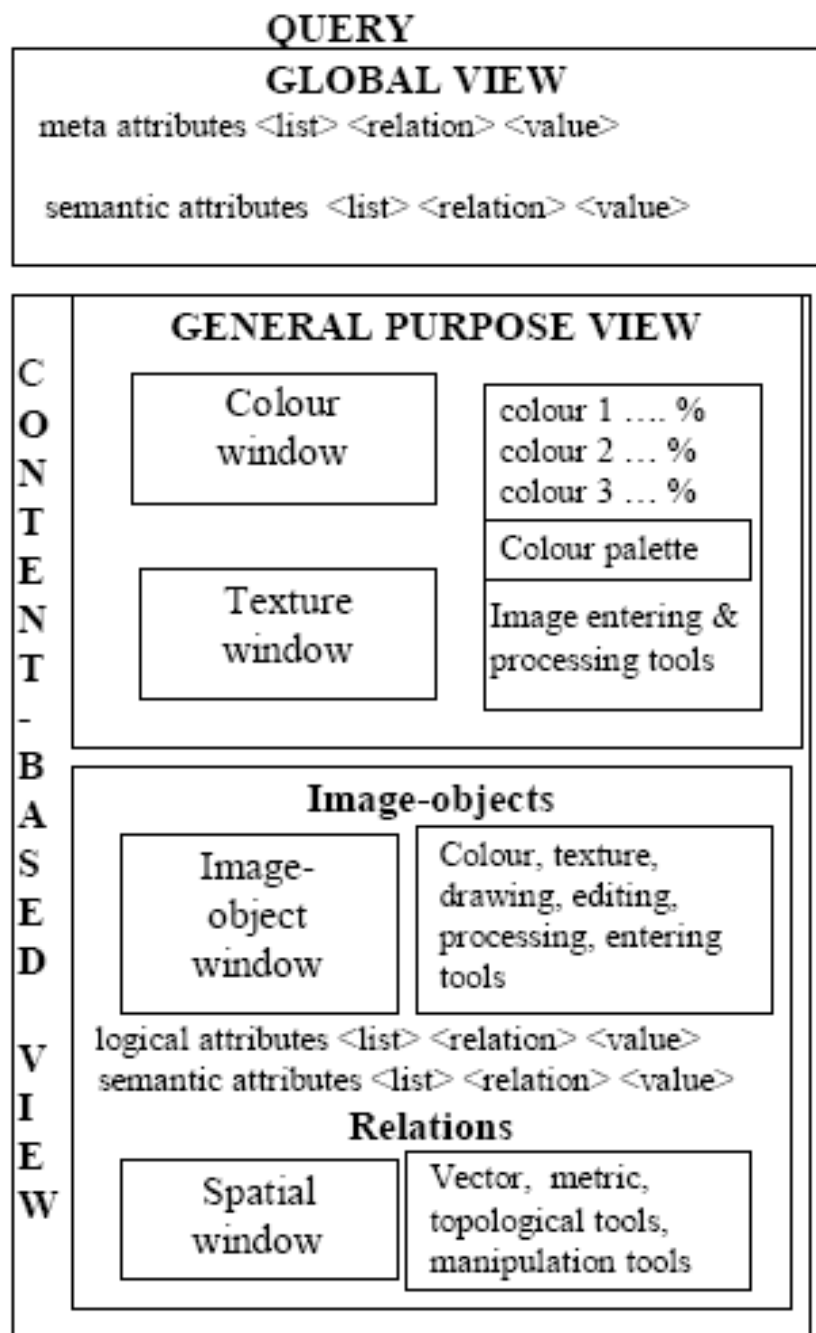
- 基于内容的图像检索（**CBIR**）
 - **CBIR**是相对于文本注释的图像检索技术而言。通过分析图像内容的底层、中层、高层特征信息来实现对示例图像或内容草图等用户提交信息的图像检索。
- 图像内容的三个层次：
 - 低层特征（**颜色**、纹理、形状）
 - 颜色直方图、颜色矩、主色调等
 - 中层特征（对象、背景、对象空间关系）
 - 高层/语义特征（场景、事件、感情等）

CBIR系统的框架



相关技术:

- 特征选择技术
- 特征提取技术
- 高维度的索引
- 相似度匹配方法
- 用户反馈的利用





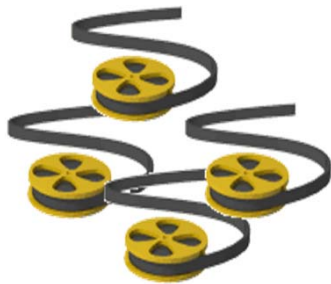


视频索引与检索

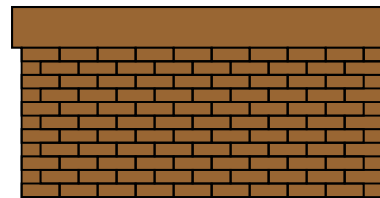
- 引言
 - 视频数据特点、用户需求
- 基于镜头的视频索引
- 镜头检测和分段
- 视频特征提取和检索
- 视频摘要和抽象

视频检索面临的问题

视频数据库

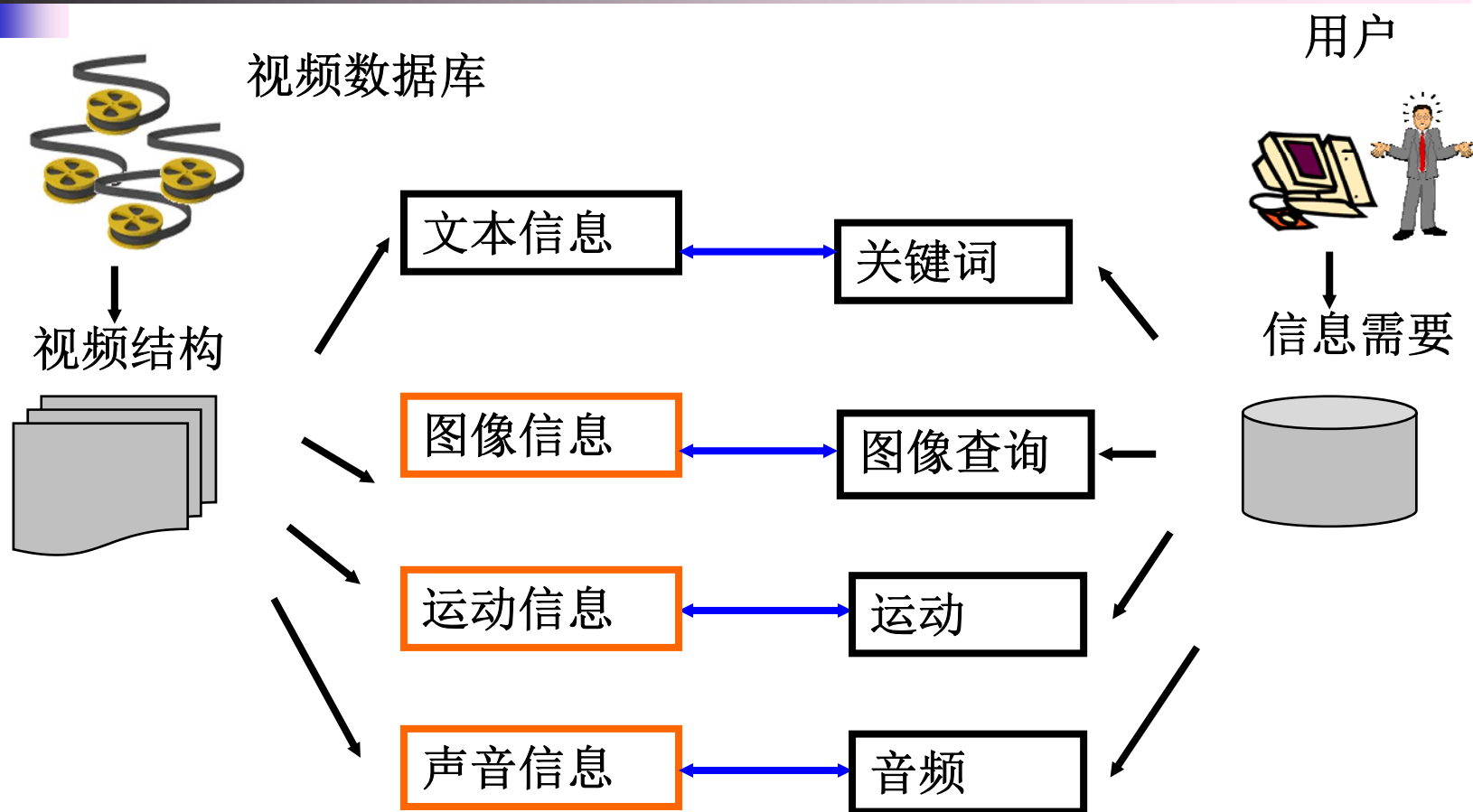


用户



结果

基于检索项的视频检索系统





2 基于镜头的视频索引

- 视频结构化过程：把连续视频流分割成包括镜头、组、场景等逻辑单元的过程。
- 视频单元：
 - 帧（Frame）：基本组成单元
 - 镜头（Shot）：基础结构层
 - 关键帧（Key Frame）：代表镜头的图像
 - 组（Group）：介于物理镜头和语义场景之间的结构
 - 场景（Scene）：由语义上相关和时间上相邻的若干镜头/组构成，描述了一个独立的故事单元（或抽象概念）

1、视频数据的基本单元
2、摄像机的一次操作所摄制的视频图像
3、代表一个场景中在时间上和空间上的连续动作



镜头切换和运动

■ 镜头切换

- **突变**切换(Cut): 一个镜头与另一个镜头之间没有过渡
- **渐变**切换: 镜头与镜头之间的变换是缓慢过渡的, 包括淡入 (fade-in) / 淡出 (fade-out)、叠化 (Dissolve)、擦除 (wipe) 等

■ 镜头运动

- 推拉镜头 (zooming)
- 摇镜头 (paning)
- 跟踪 (tracking)
- 其他



CUT镜头检测方法

■ 检测方法

- 像素层次检测 (**pixel level**)
- 块层次检测 (**block level**)
- 直方图比较 (**histogram comparison**)
- 边界跟踪 (**edge difference**)



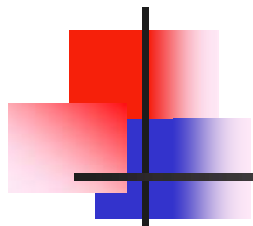
视频的特征提取和检索

- 对每个镜头加上文字或符号标识
- 代表帧（常用第一帧）特征提取与索引
- 运动特征提取与索引
- 对象特征提取与索引



视频摘要的类型

- 基于图像的
 - 视频内容切片
 - 关键帧
 - **mosaics**
- 基于视频内容的
 - 选取代表原始视频的相关片断
- 基于视频结构的
 - 场景转换图 (**scene transition graph, STG**)



全剧终!

~~谢谢各位~~