

第2章 数据库系统体系结构



一、数据库系统体系结构

- 从DBMS的角度看
 - 数据库系统内部的模式结构
- 从数据库系统的最终用户角度看
 - 数据库系统的外部体系结构

二、数据库系统的模式结构

- 从模式的角度描述一般数据库系统的概念和结构
- 可以用于解释特定数据库系统的内部结构
- **ANSI/SPARC体系结构——三级模式结构+两级映象**
 - **Oracle、MySQL等SQL数据库系统的模式结构可通过ANSI/SPARC体系结构进行解释**

1、数据库模式的概念

■ 类型（Type）和值（Value）

- 类型是指对某一类数据的结构和属性的说明
- 值是类型的一个具体赋值
- 例：类型—学生类型（学号，姓名，年龄）
值——（PB00001001，张三，20）

■ 模式（Schema）和实例（Instance）

- 模式是数据库中全体数据的逻辑结构和特征的描述，它仅仅涉及类型的描述，不涉及具体的值
- 模式的一个具体值称为模式的一个实例

2、模式和实例

- 一个模式可有很多实例
 - 模式——反映数据的结构及联系
 - 实例——反映的是某一时刻数据库的状态
- 模式相对稳定，而实例相对变动

3、模式和实例举例

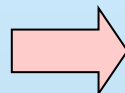
模式



- 学生表 (学号, 姓名, 年龄)
- 课程表 (课程号, 课程名, 学分)
- 选课表 (学号, 课程号, 成绩)

实际中的模式描述
比本例要详细得多

两个实例



S001	张三	21
S002	李四	20

C001	数据库	4
C002	英语	6
C003	数学	6

S001	C001	90
S002	C001	80

S001	张三	21
S002	李四	20
S003	王五	22

C001	数据库	4
C002	英语	6
C003	数学	6

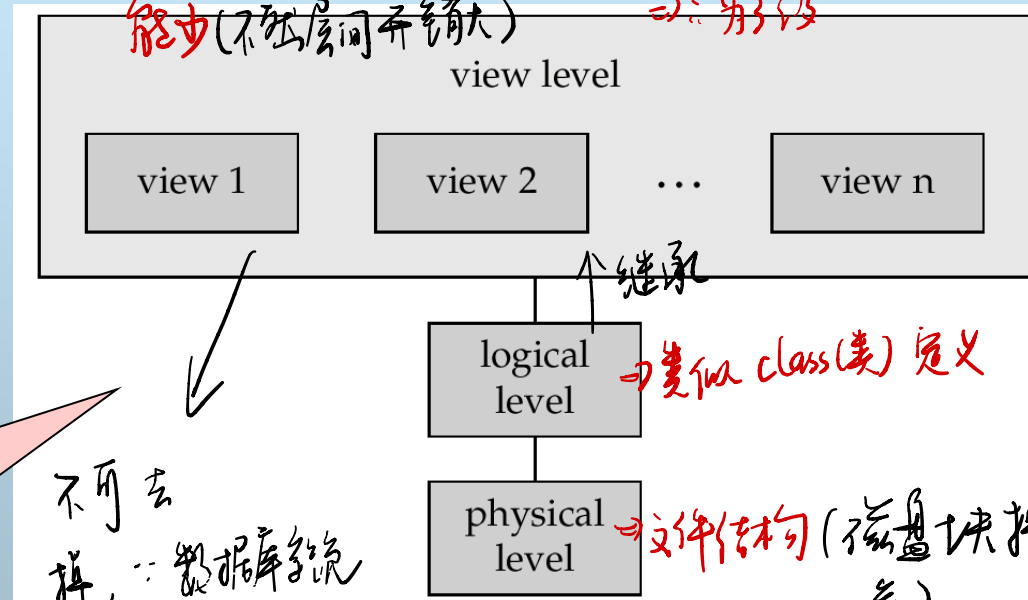
S001	C001	90
S002	C001	80
S003	C001	90
S003	C002	96
S003	C003	98

4、数据库的三级模式结构

- 外模式
- 概念模式
- 内模式

外模式是单个用户的视图，概念模式是所有用户的公共数据视图，内模式是数据库内部的存储视图

多级结构 \Rightarrow 要几级? 3级是为了防止因系统、文件系统、机器结构改变而导致数据库源码上的改动; 但又想让层级尽可能少(不然层间开销大) \Rightarrow 为3级



里是多种权限的用户,

访问需求不同, \therefore logical level 之上还需再有层抽象

现在趋势: 再多一层: 数据访问层 \Rightarrow 解决分布式存储(科大西电用mysql, 东电用oracle, 也可整合对外提供统一接口)

5、概念模式（模式、逻辑模式）

- 数据库中全体数据的逻辑结构和特征的描述
 - 数据记录由哪些数据项构成
 - 数据项的名字、类型、取值范围
 - 数据之间的联系、数据的完整性等
- 不涉及数据物理存储的细节和硬件环境
- 一个数据库只有一个概念模式
- **概念视图**：概念模式的实例 *→ 就是数据库全体数据的集合。*
- 通过**模式DDL**进行定义
 - **DDL: Data Definition Language**, 负责操作模式的数据库语言

6、外模式（子模式、用户模式）

- 单个用户所看到的局部数据的逻辑结构和特征的描述 用户有权限制自己的外模式（安全性、访问需求）
- 用户与数据库系统的数据接口，对于用户而言，外模式就是数据库
- 建立在概念模式之上，同一模式上可有多个不同的外模式
- **外部视图**：外模式的实例
- 通过**外模式DDL**进行定义

7、内模式（存储模式）

■ 数据物理结构和存储方式的描述

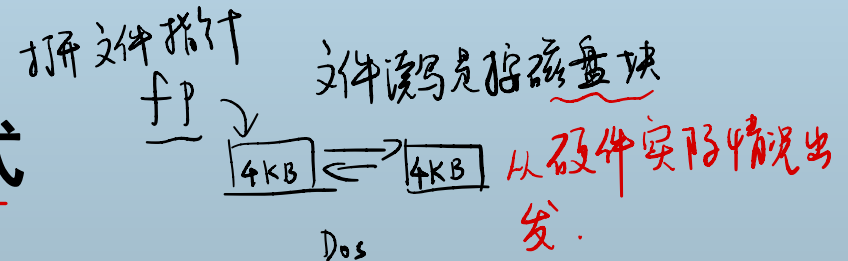
- 记录的存储方式：顺序存储、按B树组织还是散列存储？
- 索引按什么方式组织：排序、散列？
- 数据是否加密？是否压缩存储？

■ 不涉及物理块（或页）的大小，也不考虑具体设备的柱面或磁道大小

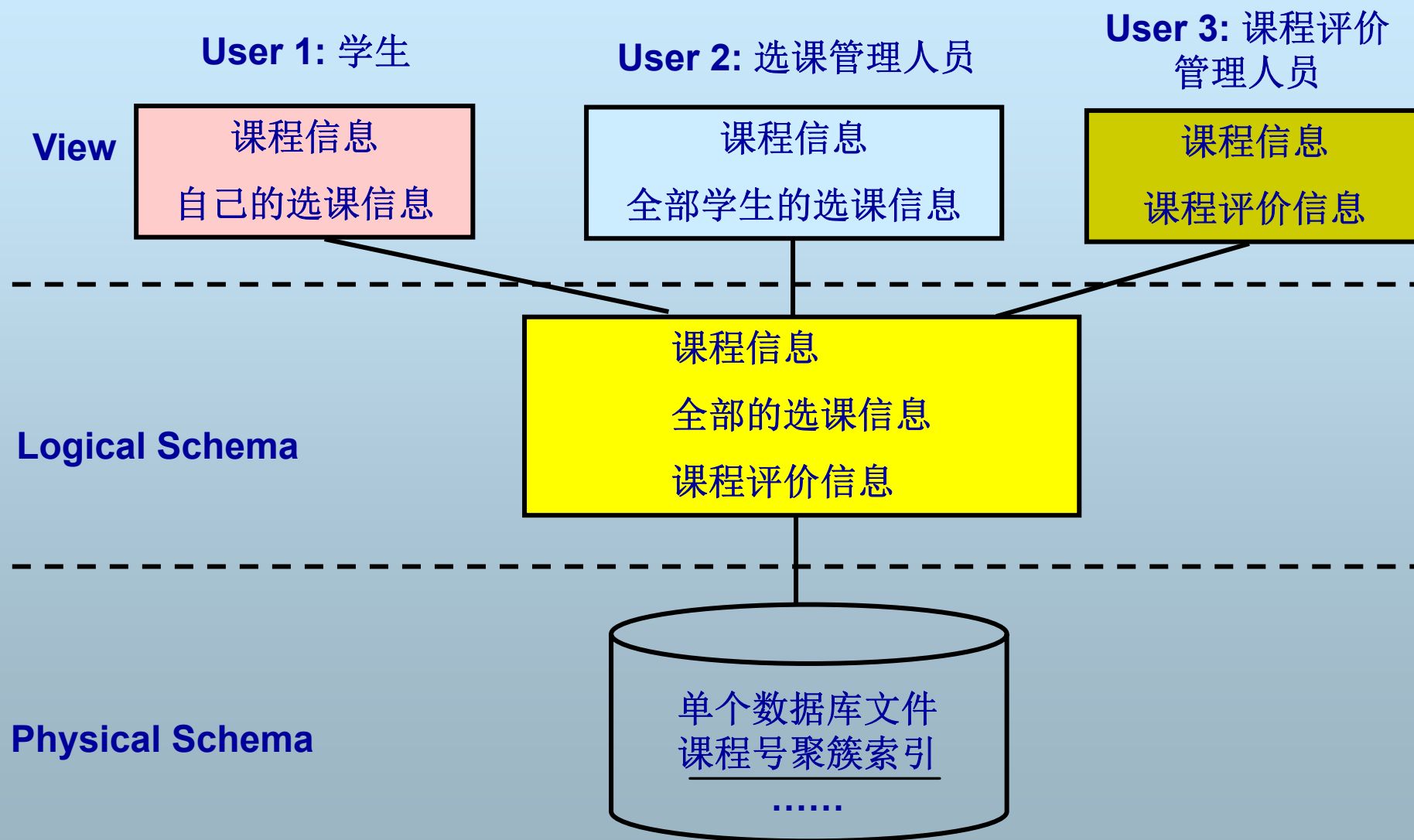
■ 一个数据库只有一个内模式

■ 内部视图：内模式的实例

■ 通过内模式DDL定义

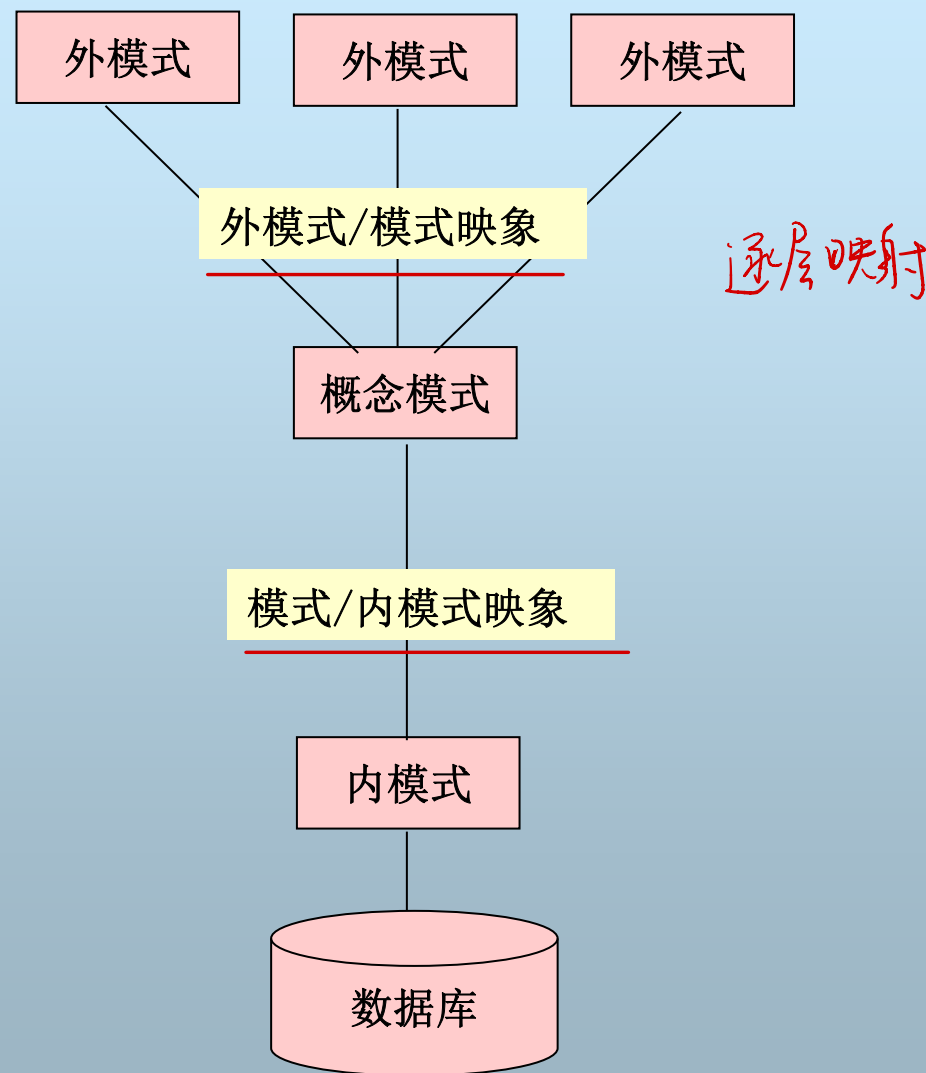


举例



三、二级映象和数据独立性

- 二级映象实现三级模式结构间的联系和转换，使用户可以逻辑地处理数据，不必关系数据的底层表示方式和存储方式



1、外模式/模式映象

- 定义了外模式与概念模式之间的对应关系
 - 属性名称可能不同
 - 外模式的属性可能由模式中的多个属性运算而得
- 当概念模式发生改变时，只要修改外模式/模式映象，可保持外模式不变，从而保持用户应用程序不变，保证了数据与用户程序的逻辑独立性
 - 数据的逻辑独立性

2、模式/内模式映像

如按学号还是姓名

排序→但新

用户的逻辑

结构不变

- 定义了概念模式与内模式之间的对应关系

- 概念模式中的逻辑记录和字段在内部如何存储

(先有序集合)

即姓名、学号

记录前

无所谓)

- 当数据库的内部存储结构发生改变时，只要修改模式/内模式映像，可保持概念模式不变，从而保持外模式以及用户程序的不变，保证了数据与程序的物理独立性

——数据的物理独立性

举例

■ 外模式：EMP (EMP, DEPT, NAME)

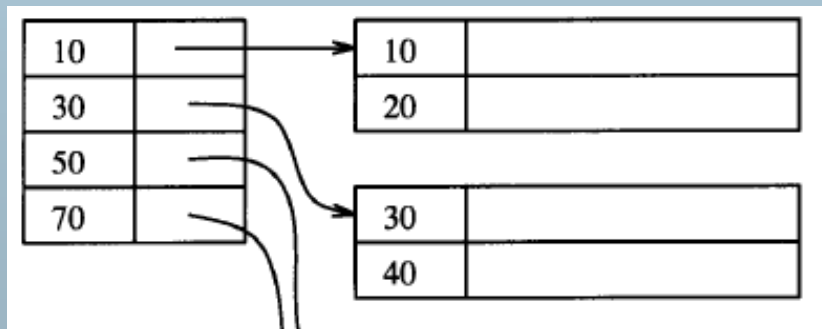
```
Create View EMP(EMP,DEPT,NAME)
As
Select E# as EMP,D# as DEPT,name
From Employee
```

此语句定义了外模式，同时也定义了外/模映象关系

E# → EMP#

■ (View/层) 模式：Employee(E#,D#,Name,Salary)

■ 内模式：顺序文件，索引文件，.....



E# → EMP#

此层改动后

只需改动
(以为逻辑
结构独立
性的例子)

```
Drop View EMP;
Create View
EMP(EMP,DEPT,NAME)
As
Select EMP# as EMP,D# as
DEPT, name
From Employee;
```

三、数据库系统外部体系结构

- 客户机/服务器结构 C/S
- 浏览器/服务器结构 B/S
- 分布式结构

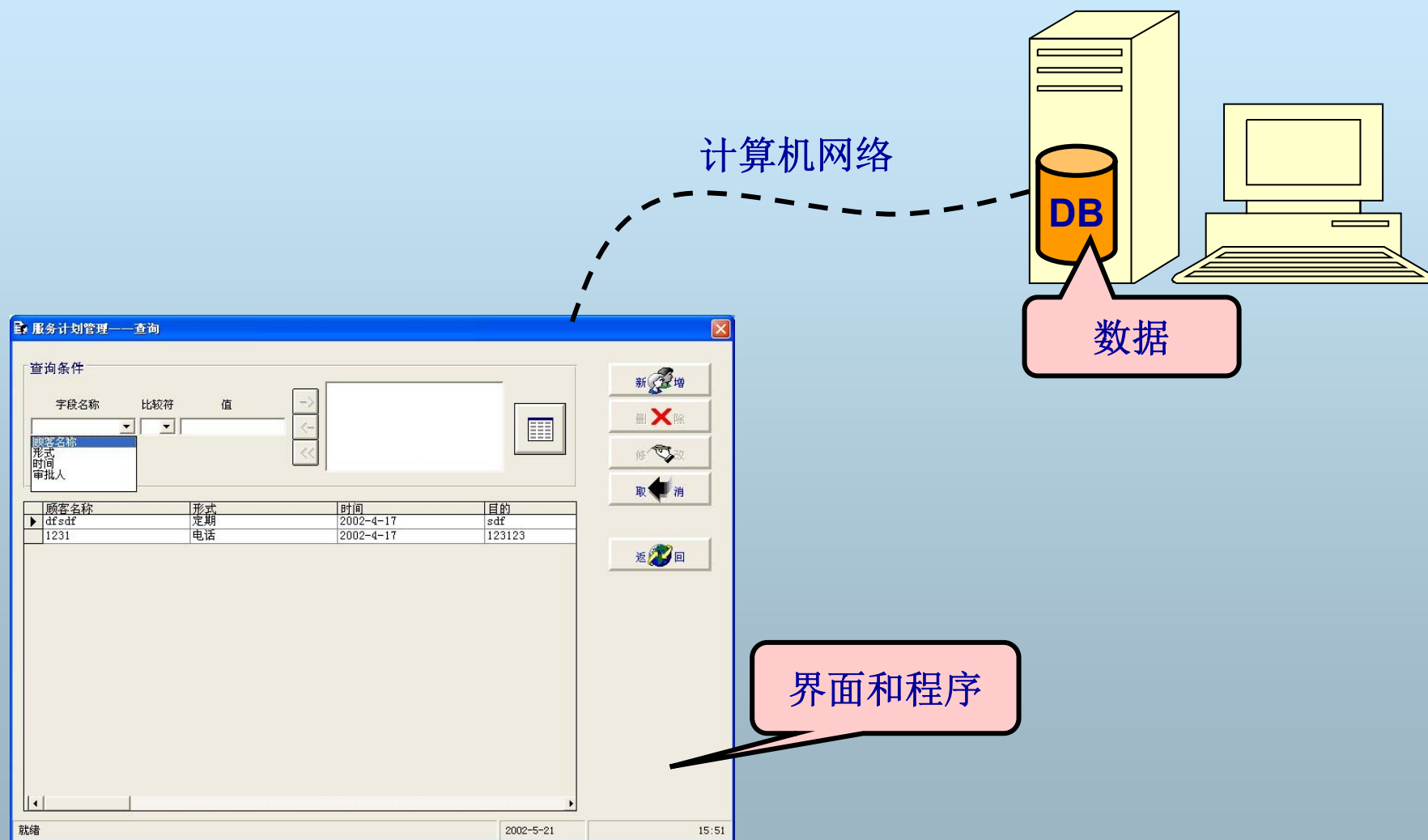
1、客户机 / 服务器结构的数据库系统

■ 把DBMS功能和应用分开

- 网络中某个（些）结点上的计算机专门用于执行DBMS功能，称为数据库服务器（Server）
- 其他结点上的计算机安装DBMS的外围应用开发工具、用户的应用系统，称为客户机（Client）

■ Client/Server架构，简称C/S架构

1、客户机 / 服务器结构的数据库系统



1、客户机 / 服务器结构的数据库系统

■ 优点

- 客户端的用户请求被传送到数据库服务器，数据库服务器进行处理后，只将结果返回给用户，从而显著减少了数据传输量
- 客户机与服务器一般都能在多种不同的硬件和软件平台上运行
- 可以使用不同厂商的数据库应用开发工具

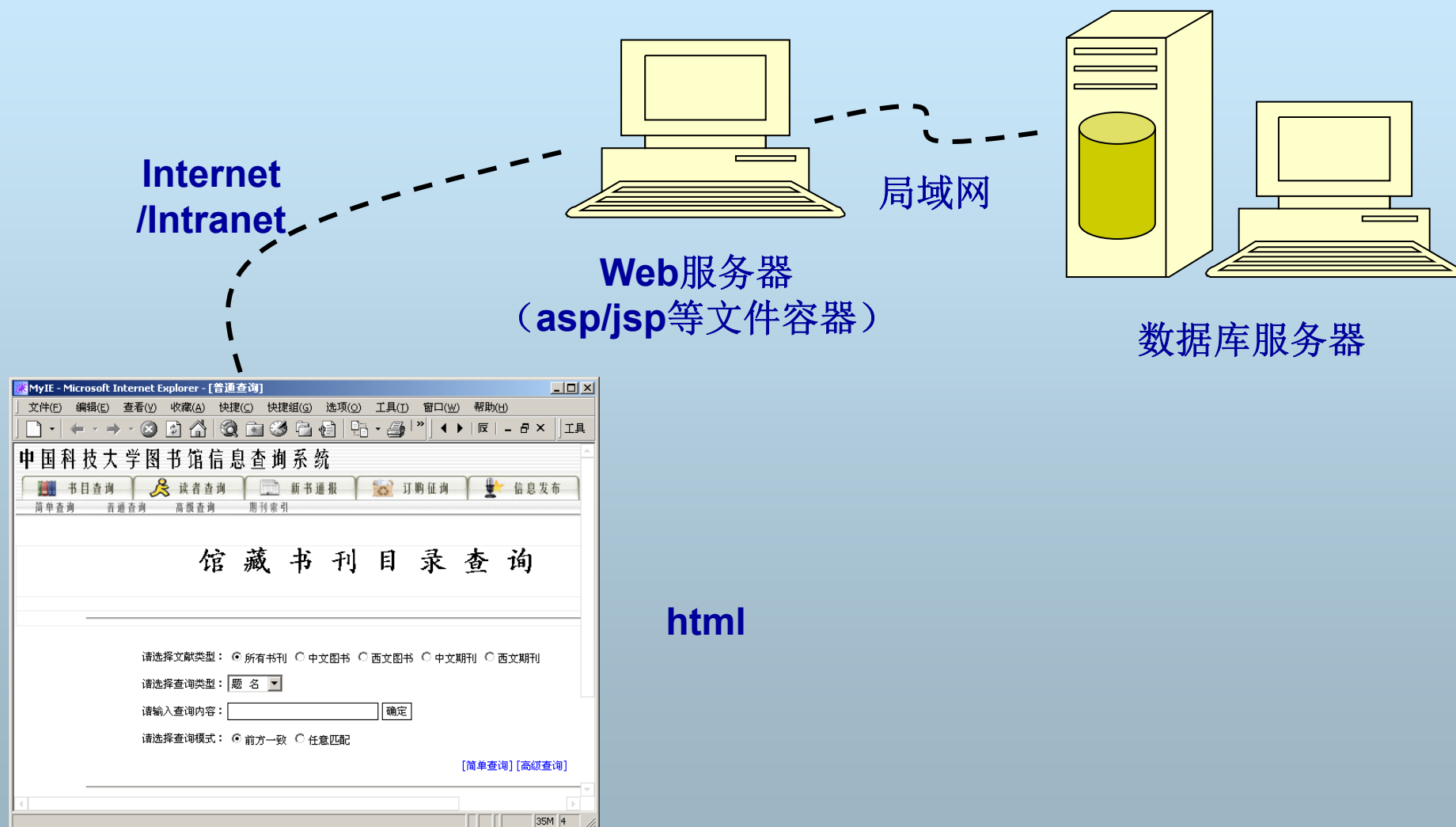
■ 缺点 (维护复杂)

- 系统安装复杂，工作量大。
- 相同的应用程序要重复安装在每一台客户机上，从系统总体来看，大大浪费了系统资源。

2、浏览器 / 服务器结构的数据库系统

- 与客户机/服务器结构类似，不同之处在于
 - 客户机统一使用浏览器软件，统一的用户界面
 - 服务器包括Web服务器和数据库服务器

2、浏览器 / 服务器结构的数据库系统



2、浏览器 / 服务器结构的数据库系统

■ 优点

- 统一的客户机界面，减少了应用安装和维护的工作量
- 基于**Web**技术，支持互联网应用

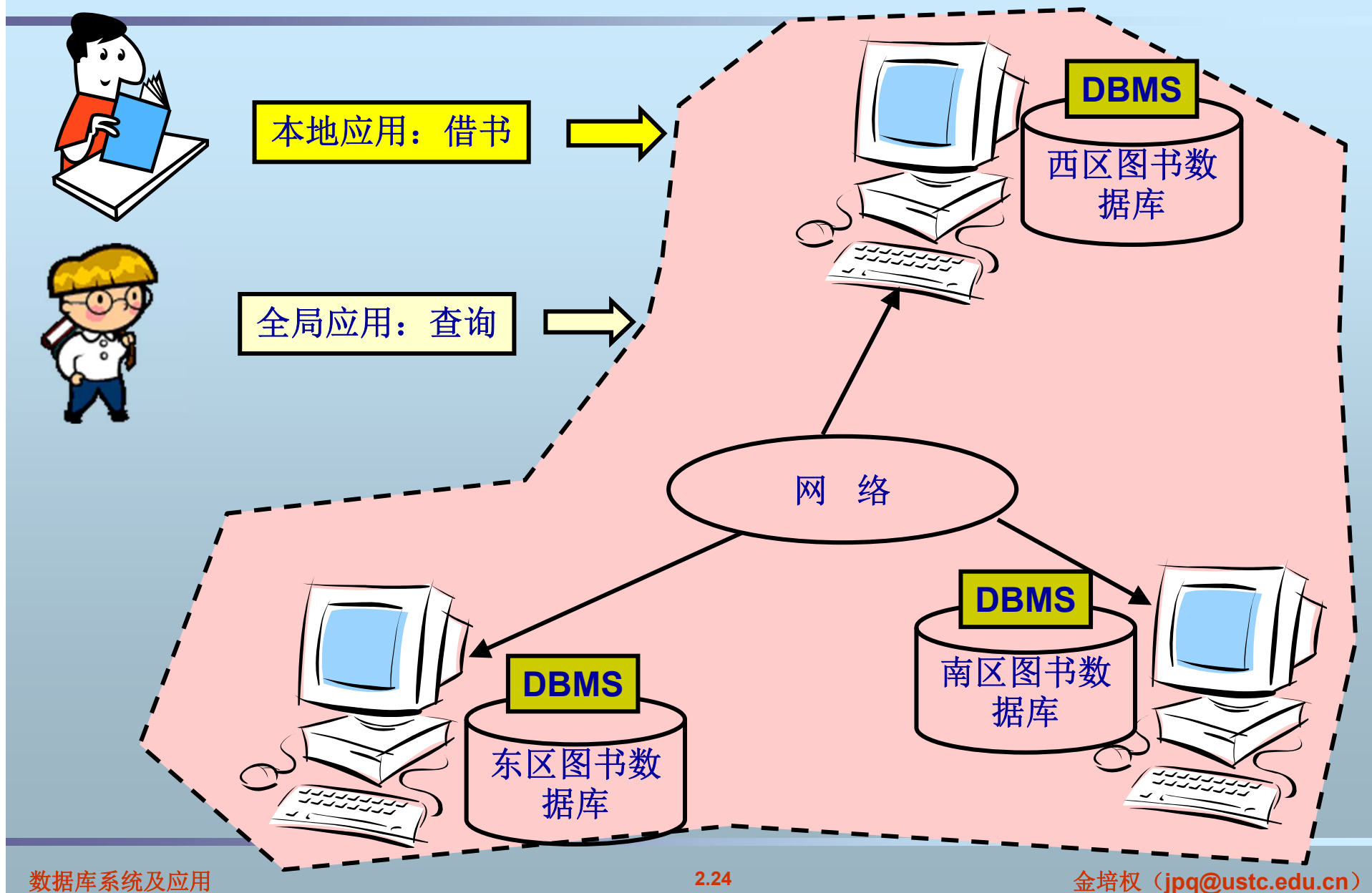
■ 缺点

- 安全性问题：用户访问无地域限制，难以控制网络恶意攻击
- 开发工具的能力相对较弱

3、分布式结构的数据库系统

- 数据库中的数据在逻辑上是一个整体，但物理地分布在计算机网络的不同结点上。
- 网络中的每个结点都可以独立处理本地数据库中的数据，执行局部应用
- 同时也可以同时存取和处理多个异地数据库中的数据，执行全局应用
- 特点
 - 物理分布、逻辑统一

3、分布式结构的数据库系统



3、分布式结构的数据库系统

■ 优点

- 适应了地理上分散的公司、团体和组织对于数据库应用的需求。

■ 缺点

- 数据的分布存放给数据的处理、管理与维护带来困难。
- 当用户需要经常访问远程数据时，系统效率会明显地受到网络传输的制约。

本章小结

- 模式和实例的概念
- 三级模式结构
- 二级映像
- 数据独立性
- 数据库系统外部体系结构