

课程名称: 数据库系统

关系模型

数据如何存储

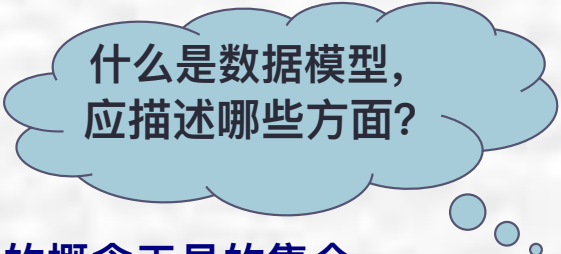
- 文件，无序，无结构
- **EXCEL**
- 数据库，二维表

主要学习目标

- 关系模型的数据结构
- 关系模型中的完整性约束

一 数据模型

1.1 数据模型的三大基本要素



什么是数据模型，
应描述哪些方面？

- 数据模型：

是一个描述数据、数据联系、数据语义以及数据一致性约束的概念工具的集合

- 应包括：

- 数据结构：

由一组创建数据库的规则（定义数据库的结构）组成

- 数据操作：

定义对数据进行的操作类型（包括**更新和查找**数据库中的数据以及**修改**数据库的结构）

- 约束条件：

一组数据完整性定义规则，确保数据的正确性。



前测小问题

- 概念模型设计的结果是什么？
- 逻辑设计的目的是什么？

1.2 层次(数据)模型

- 第一个商用的层次数据模型系统：
IBM公司开发的数据库管理系统**IMS**
(Information Management System)
- 是层次数据模型典型代表
- 曾经广泛使用

(一)层次模型的基本概念

- 层次模型

利用“**记录**”(包含多个“**属性**”)和“**双亲子女关系(PCR)**”来描述应用的数据结构

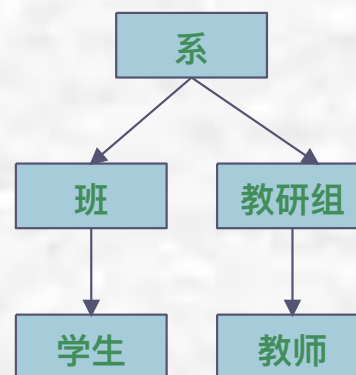
- 层次模式型)

利用层次模型描述一个应用的数据结构，称为一个层次模式（型：数据库的结构），为“**树**”结构。

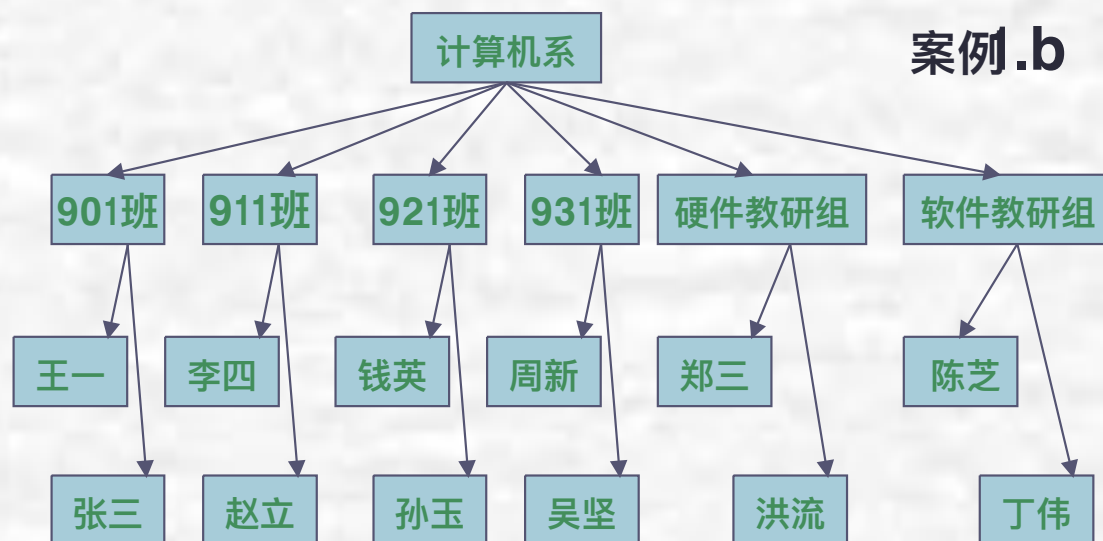
- 层次模式的**实例**

一个型有多个实例(数据库的数据)
为“**森林**”结构

层次模型如何描述数据的结构？



案例.a

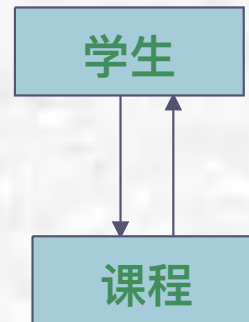


案例.b

(二) 非层次结构的描述

- 问题

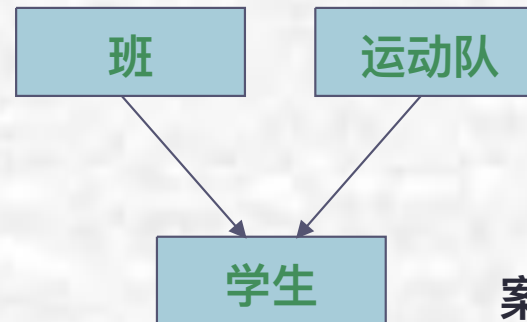
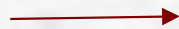
M:N联系



案例.c

* 非层次的数据结构也可以描述吗?

多双亲联系



案例.d

多数客观数据都不具有层次关系!)

1.2 层次模型

(二) 非层次结构的描述

- 解决方法一：采用副本

对M：N联系

案例.e



对多双亲联系

案例.f



缺点：数据冗余（增加空间，一致性维护难）

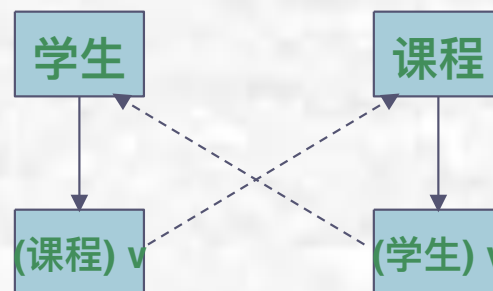
* 如何避免这种现象?

1.2 层次模型

(二) 非层次结构的描述

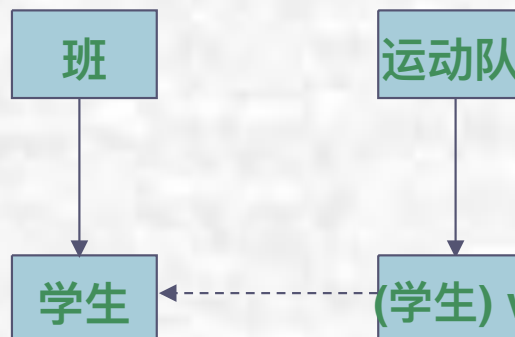
- 解决方法二：虚拟记录(优化方法)

对M：N联系



案例.g

对多双亲联系



案例.h

不足：指针操作增加开销

1.2 层次模型

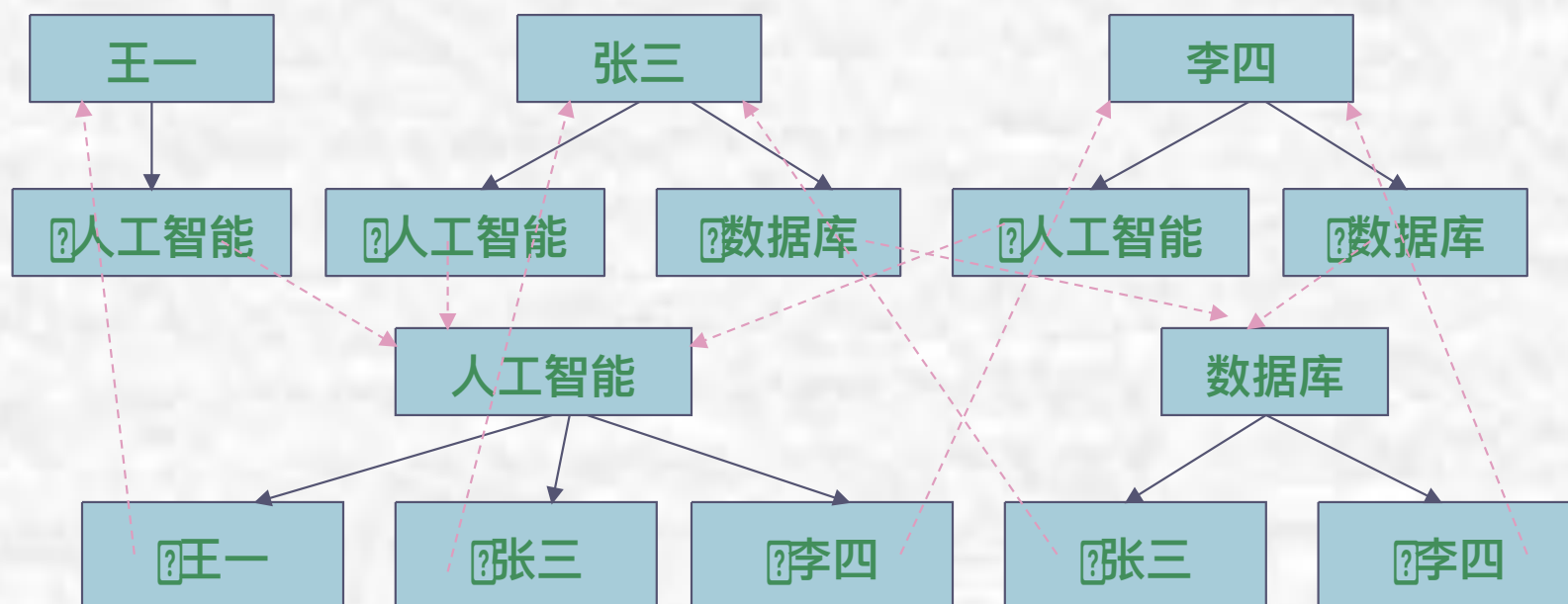
(二) 非层次结构的描述

- 非层次联系模式的实例

M: N联系举例

虚拟记录方式

* 非层次结构的数据库实例什么样?



注: ? - 表示是虚拟记录 (地址指针)

案例.i

(三) 定义物理存储结构

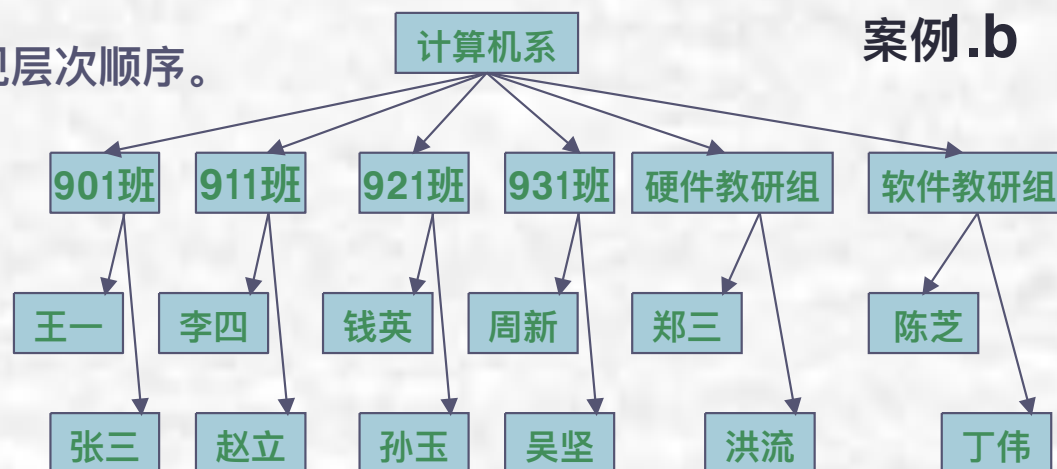
- 邻接法:

按照层次树前序穿越的顺序
，把所有记录值(子段定长, **定长记录**
)依次邻接存放。

即通过物理空间的位置相邻来实现层次顺序。

* 层次模式数据,
物理上如何存放?

- 例示:



计算机系	901班	王一	张三	911班	李四	吴坚	921班	...	硬件教研组	郑三	洪流	软件教研组	陈芝	丁伟
------	------	----	----	------	----	----	------	-----	-------	----	----	-------	----	----

案例.j

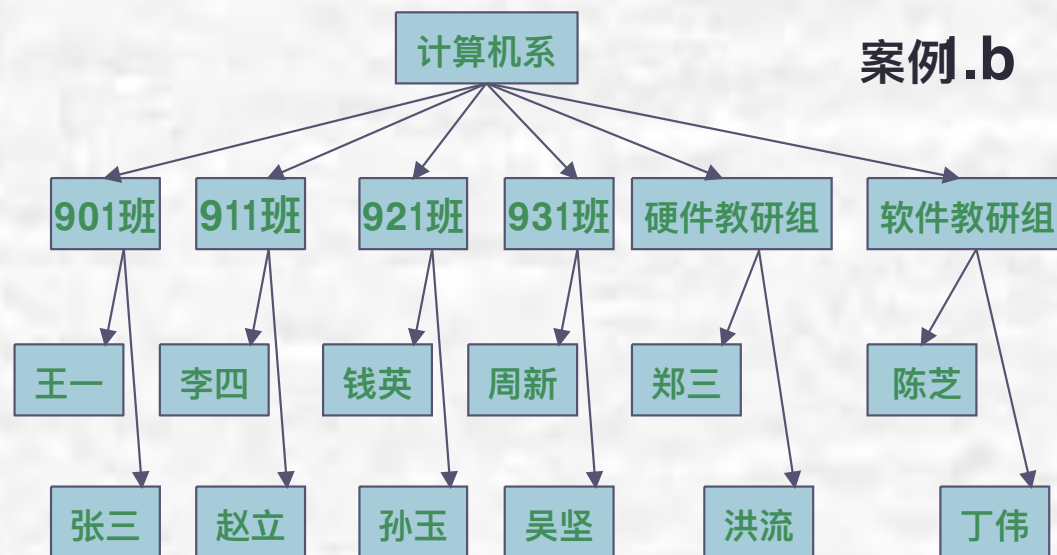
1.2 层次模型

(四) 定义基本数据操作

* 层次模型，应提供什么数据操作？

查找一个记录：从根记录开始，按给定条件沿层次路径进行搜索。

- Get unique(GU)
查找给定条件的记录
- Get next within parent(GNP)
查找下一亲兄弟
- Get next(GN)
找到当前记录的下一记录
-



GU 系(系名 = ‘计算机系’), 班(班名 = ‘911 班’), 学生;

while not fail do GNP 学生; /* 找出 911 班所有学生记录 */

GN 学生; /* 找到 921 班的第一个学生记录, 即钱英 */

while not fail do GNP 学生; /* 找出 921 班所有学生记录 */

*(五) 层次模型小结

优点：

- 数据模型比较简单，操作简单。
- 对于实体间联系是固定的，且预先定义好的应用系统，性能较高。
- 提供良好的完整性支持。

缺点：

- 不适合于表示非层次性的联系。
- 对插入和删除操作的限制比较多。
- 查询子女结点必须通过双亲结点。

1.3 网状(数据)模型

- 第一个网状数据模型系统:

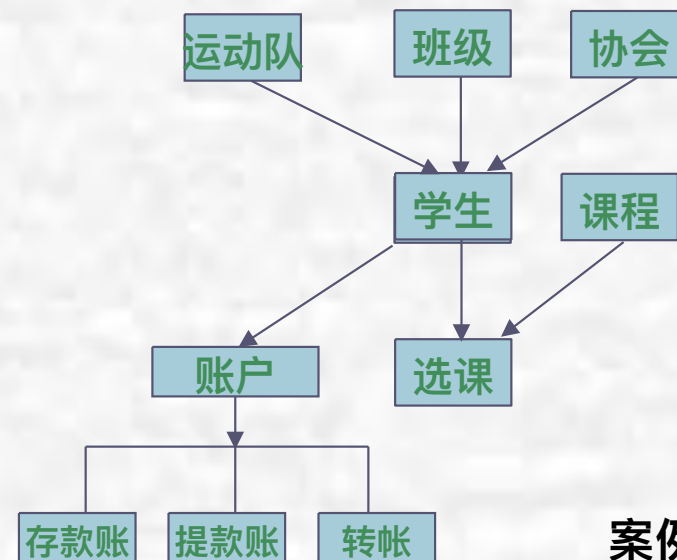
美国通用电器公司（Bachman等人）开发的数据库管理系统**IDS**
(Integrated Data Store)

- 奠定了网状数据库的基础
- 曾经**广泛应用**

(二) 网状模式与实例

* 网状模型的模式与实例?

- 网状模式（型）
 - 利用网状模型（记录&系）来描述一个应用，可以得到一个网状模式（数据库的结构）
 - 是一个“图”（而非一个“树”）



案例2.c

- 实例
 - 一个型□多个实例(数据库的数据)

(一) 网状模型的基本概念

网状模型如何描述数据的结构？

- 记录的表示(内部结构)
记录数据项 (允许为多值和复合数据)
- 联系的表示(外部结构)

系 (set)

单属系:

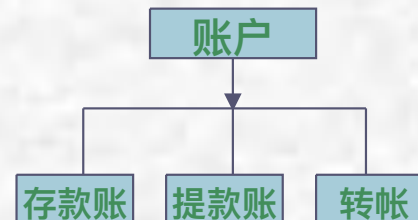
多属系:

首记录&属记录,

属记录值可以是不同不同记录类型



案例2.a



案例2.b

- 网状模型的不同特点
 - 它去掉了层次模型的两个限制:
 - 允许多个结点没有双亲结点;
 - 允许结点有多个双亲结点。
 - 它还允许两个结点之间有多种联系 (复合联系)

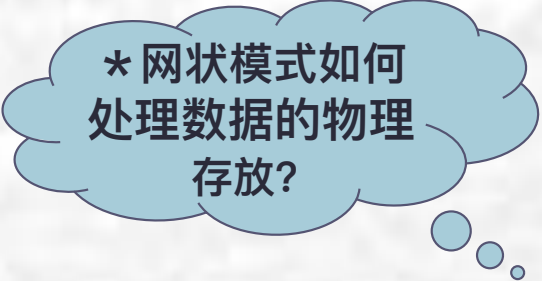
(三) 定义物理存储结构

- 网状数据模型的存储结构

依具体系统不同而不同，常用的方法是链接法，包括

- 单向链接
- 双向链接
- 环状链接
- 向首链拉等。

- 示例（见下页）



* 网状模式如何处理数据的物理存放？

2.1 网状模型

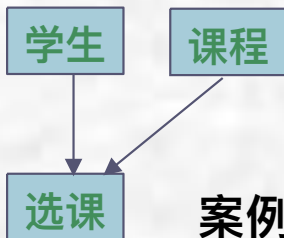
物理存储结构的一个示例

案例2.e

(学号,姓名,系别)

(课号,课程名,学分)

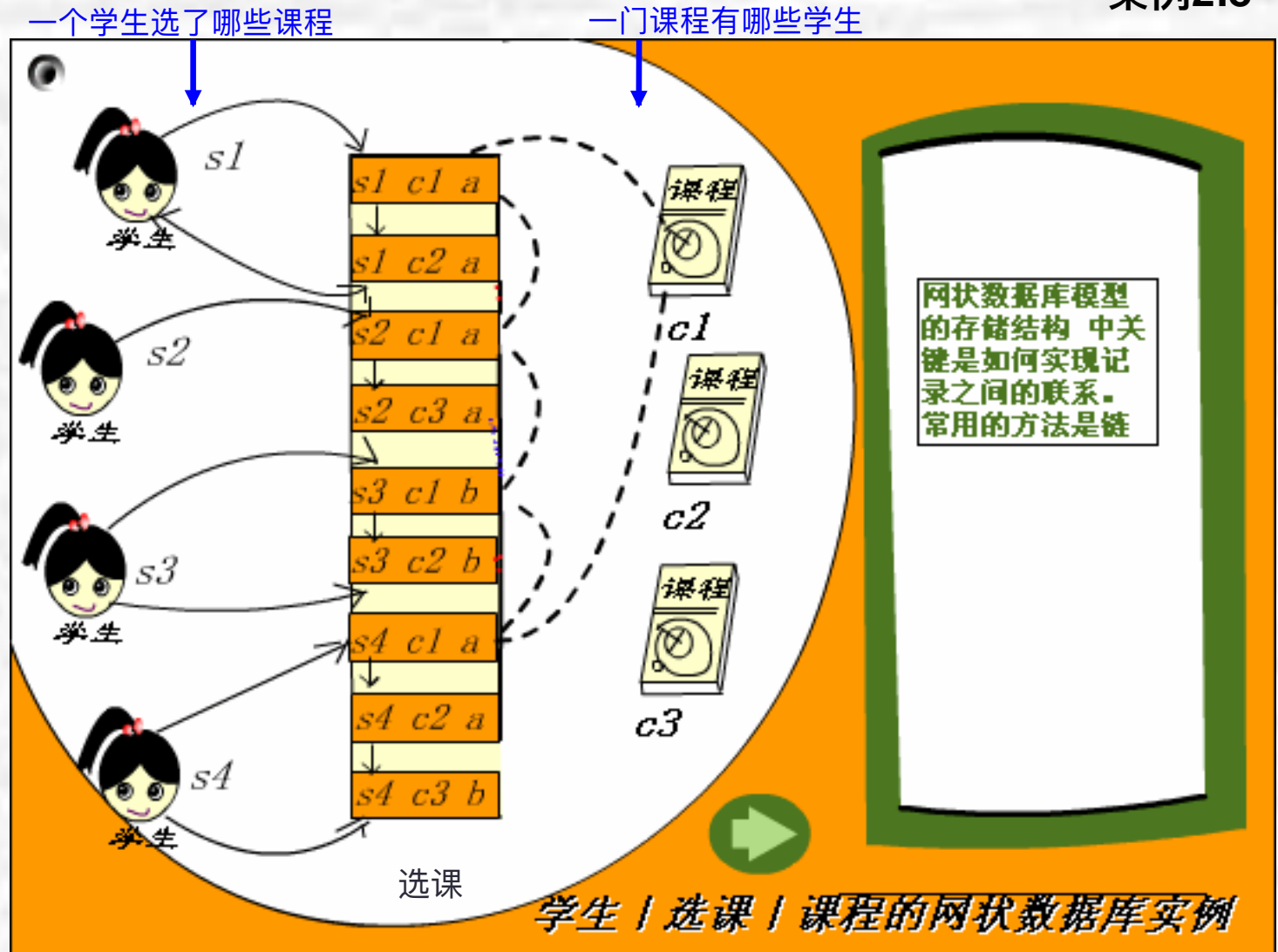
(学号,课程号,成绩)



案例2.d

* 网状模型(的属性)为何允许复杂数据类型?

因为物理实现
采用指针链!



(四) 定义基本的数据操作

- Find(查找)
- Get(取数)
- Store(存数)
- Modify(修改)
- Erase(删除)
- Connect(加入)
- Reconnect(转接)
- Disconnect(撤离)

*（五）网状模型小结

优点：

- 能够更为直接地描述现实世界。
- 具有良好的性能，存取效率较高。

缺点：

- 其DDL语言极其复杂。
-

数据独立性较差。由于实体间的联系本质上通过存取路径指示的，因此应用程序在访问数据时要指定存取路径。

二 关系模型的基本概念

2.1 定义数据对象的结构

关系模型如何描述
一个数据对象的(
内部)结构?

元组→

属性

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

关系模式

关系
(实例)

案例3.a *instructor* 关系 ← 关系名

采用关系(table)描述:
关系名加上一组属性!

关系模式与实例指什么?
元组、属性和关系又指什么?

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

案例3.b *course* 关系

2.2 定义数据对象间的关联

关系模型如何描述
数据对象间的关联
关系?

仍采用关系(table)描述:
关系名加上一组属性!

<u>course_id</u>	<u>prereq_id</u>
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

案例3.c *prereq* 关系

例如:

- 1)课程间的先修关系
- 2)学生与课程间选课关系
- 3)教师与学生间授课关系
- 4)教师, 课程, 学生间关系

<u>course_id</u>	<u>title</u>	<u>dept_name</u>	<u>credits</u>
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

案例3.b *course* 关系

如何理解属性的原子特性？

2.3 第1范式(1NF)

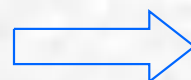
要求关系的属性具有原子性！

• 纵向展开问题

教师编号	教师名	电话号码
1001	张三	65102502, 65102003, 13231243124
1002	李四	65102502, 13423431788

案例3.d 教师联系方式

分量个数不定



教师编号	教师名	电话号码
1001	张三	65102502
1001	张三	65102003
1001	张三	13231243124
1002	李四	65102502
1002	李四	13423431788

案例3.f

• 横向展开问题

教师编号	教师名	住址
1001	张三	重庆市沙坪坝区沙正街174#
1002	李四	重庆市沙坪坝区劳动路1#

案例3.e 教师住址

分量个数固定



教师编号	教师名	省市	区县	街道	门牌号
1001	张三	重庆市	沙坪坝区	沙正街	174#
1002	李四	重庆市	沙坪坝区	劳动路	1#

案例3.g

不希望的方式

希望的方式(信息好用)

三码与模式图

3.1 主码完整性约束

什么是关系的主码(键)、超码和候选码，各自特点和作用？

该关系的特点？

具有多(4)个候选码，多个超码！

教师编号	教师名	身份证号	护照号	医保卡
1001	张三	201202199009090371	G10221466	S231456756655656
1002	李四	201202199091212370	G34431244	S675697031277755

案例4.a 教师基本信息

但一个关系仅允许有唯一的主码，区分不同元组！

```
classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)
```

案例4.c 大学数据库模式

<u>course_id</u>	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

案例4.b course 关系

<u>ID</u>	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

案例4.d instructor 关系

3.2 外码参照完整性约束

外键 (完整性约束) p.24

course_id	sec_id	semester	year	building	room number	time slot_id
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

课程安排

案例4.e section 关系

什么是关系的外键，主要作用？

作用：

- 1)说明元组间联系
- 2)保证数据有效性

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

教师 案例4.d instructor 关系

图2-7描述的数据任课关系，还需要哪些关系来支撑？

描述的数据关联：哪个教师承担什么门课！

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

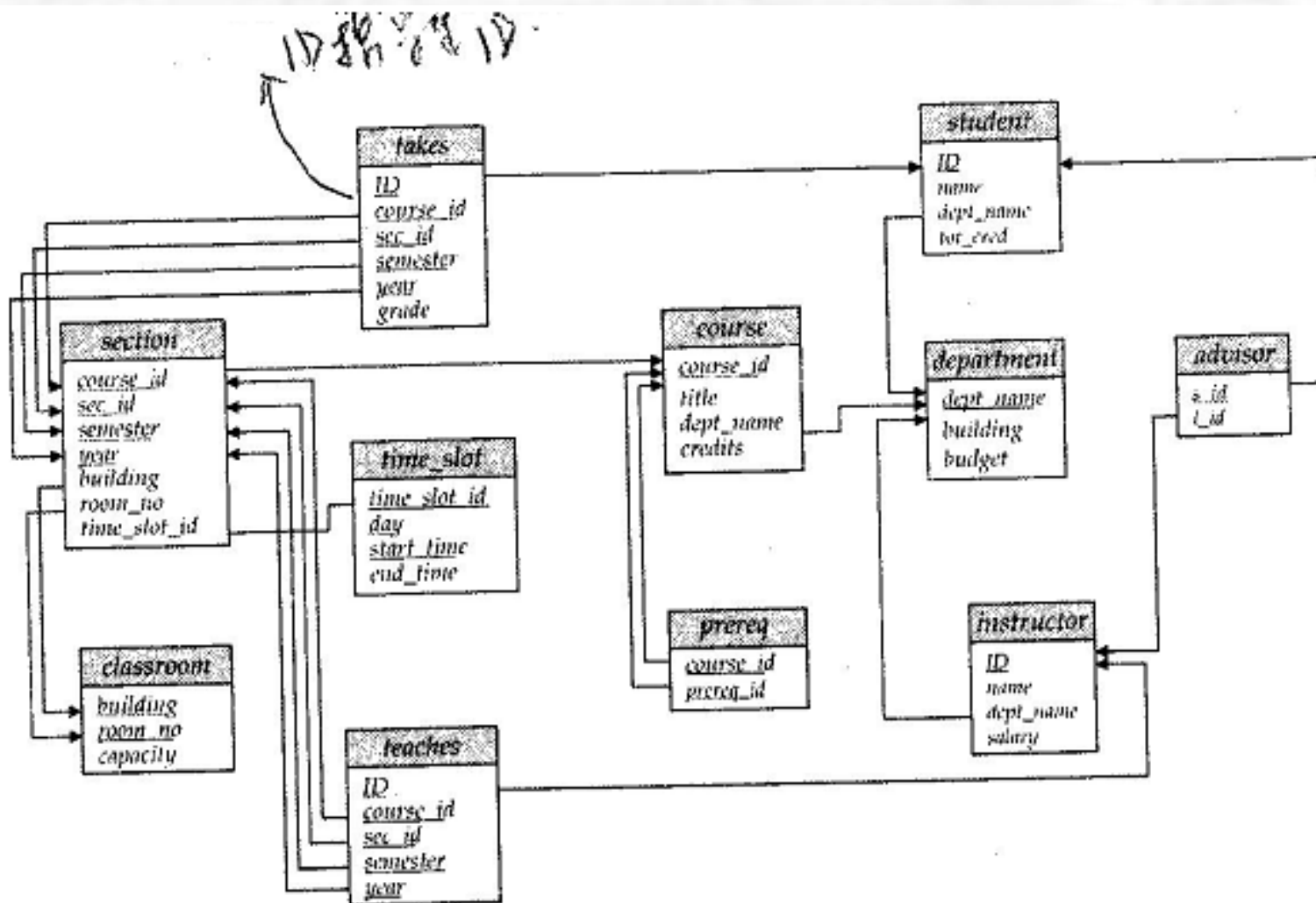
任课 案例4.f teaches 关系

3.3 模式图

模式图的主要作用?

- 数据模型
描述数据对象的
“内部结构”
“相互关联”
- 线条
外键- 参照关系,
描述对象间的关联
（“外部结构”）
- 矩形
关系模式,
描述对象的特征
（内部结构）
上方- 关系名称
下划线(组合)- 主键

(要求学生都会画)



案例5 大学数据库的模式图

四 关系模型的完整性约束

? 定义

是对关系的某种约束条件。

? 目的

用于保证关系数据库中数据的正确性和可靠性。

? 类型

? 实体完整性规则

? 参照完整性规则（引用完整性规则）

? 域完整性规则（用户自定义完整性规则）

- 语法约束
- 语义约束

4.1 实体完整性规则

- 规则要求：
 - 在任何关系的任何一个元组中，主键的值不能为空值、也不能取重复的值。
- 目的：用于保证数据库表中的每一个元组都是惟一的。

请思考：下面的关系是否违反实体完整性规则？

学号	姓名	性别	年龄	系编号
03001	马力刚	男	21	01
03102	王萍华	女	20	02
03223	王平	男	21	03
	张华	男	22	04
03001	李萍	女	19	05

4.2域完整性规则(用户定义完整性规则)

- 规则要求

- 由用户根据实际情况，定义表中属性的取值范围
 - 例如：性别只能是男和女、年龄不能为负值、成绩在0—100之间等。

- 目的



用于保证给定字段中数据的有效性,即保证数据的取值在有效的范围内。

4.3 参照完整性规则（引用完整性规则）

- 规则要求：
 - “不引用不存在的实体”
。即：不允许在一个关系中引用另一个关系中不存在的元组。
- 目的
 - 用于确保相关联的表间的数据保持一致。

4.3 参照完整性规则（引用完整性规则）

请思考：下面两个关系是否违反参照完整性规则？

系表（主表）			学生表（从表）	
系编号	系名	系主任	办公室	电话
01	计算机	龚小勇	205	6003
02	通信	谭中华	207	6025
03	电子	袁 勇	210	6018

学号	姓名	性别	年龄	系编号
03001	马力刚	男	21	01
03102	王萍华	女	20	
03223	王平	男	21	05

- 说明

- 从表的“系编号（外键）”的取值只能为两种情况：

- 若取非空值，则它必须是主表中**存在的值**。

-

取空值(null)。表明尚未给学生分配专业。Null不等于0或空字符串。

五 关系运算

5.1 基本关系运算

应有哪些关系运算，各自作用？

常用的操作方式：

- 取关系的某些行
- 取关系的某些列
- 从多个关系取

符号(名字)	使用示例
σ (选择)	$\sigma_{\text{salary} \geq 85000}(\text{instructor})$ 返回输入关系中满足谓词的行
Π (投影)	$\Pi_{ID, salary}(\text{instructor})$ 对输入关系的所有行输出指定的属性。从输出中去除重复元组
\bowtie (自然连接)	$\text{instructor} \bowtie \text{department}$ 从两个输入关系中输出这样的元组对：它们在具有相同名字的所有属性上取值相同
\times (笛卡儿积)	$\text{instructor} \times \text{department}$ 从两个输入关系中输出所有的元组对(无论它们在共同属性上的取值是否相同)
\cup (并)	$\Pi_{name}(\text{instructor}) \cup \Pi_{name}(\text{student})$ 输出两个输入关系中元组的并
$-$ (差)	输出在第1个关系而不在第2个关系中的元组
ρ (更名操作)	

5.2 关系运算的示例

五 关系运算

dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

案例6.a department 关系

这些操作的结果是什么，有无限制条件？

选择: $\sigma_{\text{salary} > 85000}(\text{instructor})$

投影: $\Pi_{\text{ID}, \text{salary}}(\text{instructor})$

并: $\Pi_{\text{name}}(\text{instructor}) \cup \Pi_{\text{name}}(\text{student})$

差: $\sigma_{\text{salary} > 85000}(\text{instructor}) - \sigma_{\text{salary} > 92000}(\text{instructor})$

差异 { 笛卡儿积: $\text{instructor} \times \text{department}$
自然连接: $\text{instructor} \bowtie \text{department}$

(连接属性不从复，元组数更少，连接元组有意义)

更名: $\rho_x(A_1, a_2, \dots, a_n)(E)$

$\Pi_{\text{instructor.salary}}(\sigma_{\text{instructor.salary} > d.\text{salary}}(\text{instructor} \times \rho_d(\text{instructor})))$

基本运算可以组合吗？

运算结果是关系，可反复组合，操作功能增强！

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

案例6.b instructor 关系

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

案例6.c student 关系





课程总结与作业安排

- 基本知识：
 - 域、关系、关系模式
 - 关键字、主键、外键
 - 实体完整性、参照完整性、用户自定义完整性
- 扩展学习：
 - 开发软件系统的一种思维？
- 作业
 - 第2章：2.1, 2.9, 2.10