

第二章 关系模型介绍

关系模型是当前商业数据处理应用中最主要的数据库模型。

主要内容

- 关系库结构
- 数据库模式
- 码/键
- 模式图
- 关系查询语言

本章要掌握的内容

□ 关系模型的三要素

■ 数据结构

- 关系表的特点

- 优点和缺点

■ 数据操作

- 基本CRUD操作

■ 数据约束

- 码的种类

- 码的特点

- 码的作用

2.1 关系数据库的结构

□ 一个关系型数据库包括：

- 表/关系的集合：每个表有唯一的名称

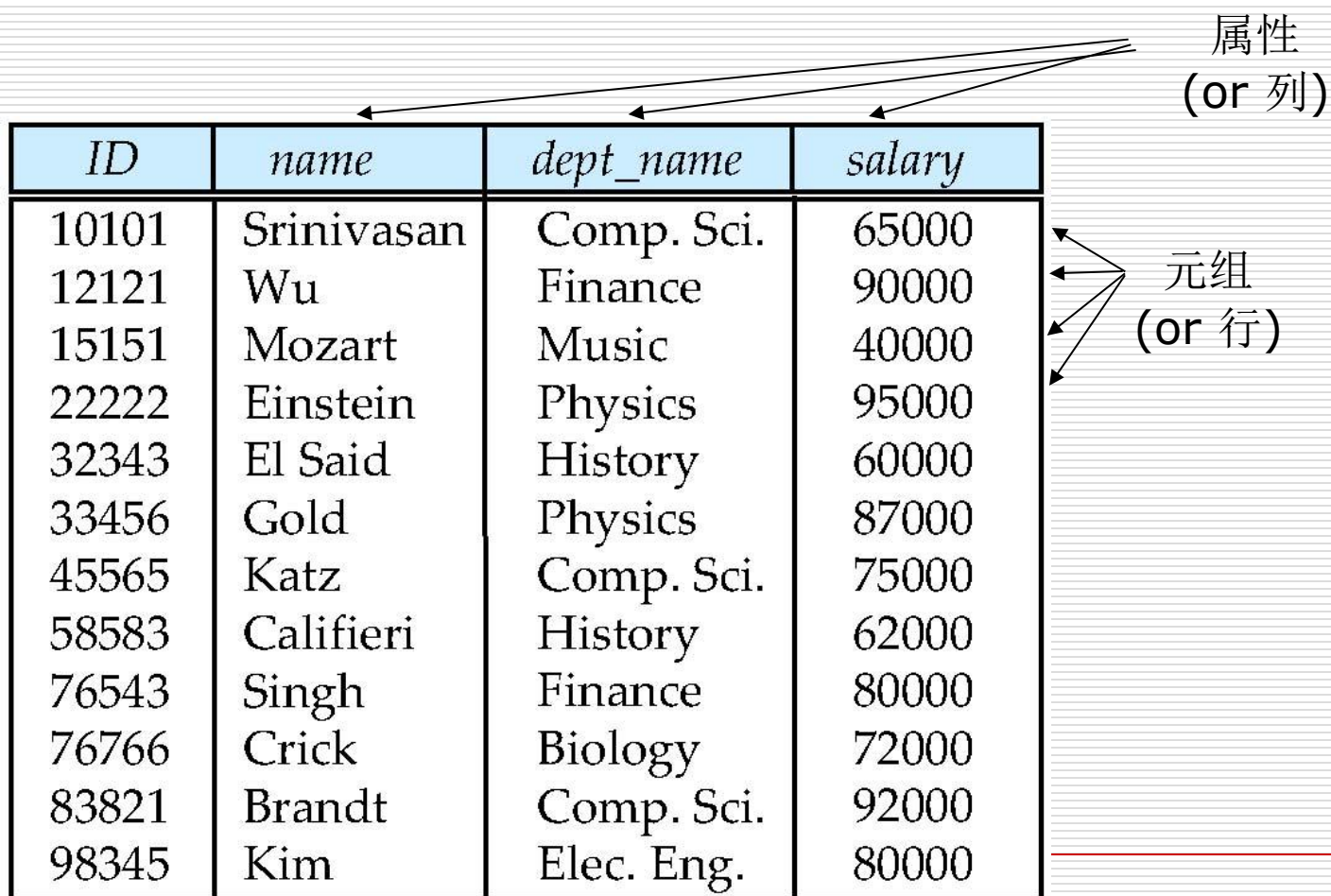
□ 列/属性的集合

- 列名
- 域：属性被允许的取值范围
- 原子性
- Null:空值

□ 行/元组的集合

- 表示了一个关系的取值的集合
 - 一行代表了一组值之间的某种联系
-

关系表举例



The diagram shows a table with four columns: *ID*, *name*, *dept_name*, and *salary*. The first row of data is highlighted. Annotations with arrows point to the columns from the text '属性 (or 列)' and to the first row of data from the text '元组 (or 行)'.

| <i>ID</i> | <i>name</i> | <i>dept_name</i> | <i>salary</i> |
|-----------|-------------|------------------|---------------|
| 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 |

关系的相关术语

□ A_1, A_2, \dots, A_n 是属性

□ $R = (A_1, A_2, \dots, A_n)$ 是关系模式

如: $instructor = (ID, name, dept_name, salary)$

□ $r(R)$ 表示关系模式R上的关系

□ 一个关系的当前值（关系实例）由一个表确定

□ r 的一个元素 t 是一个元组 *tuple* , 用表 *table*中的一行 *row*表示

关系是无序的: Unordered

- 元组的顺序是不相干的irrelevant (元组可以任意顺序存储)
- 例如: 具有无序元组的关系*account*

| <i>account_number</i> | <i>branch_name</i> | <i>balance</i> |
|-----------------------|--------------------|----------------|
| A-101 | Downtown | 500 |
| A-215 | Mianus | 700 |
| A-102 | Perryridge | 400 |
| A-305 | Round Hill | 350 |
| A-201 | Brighton | 900 |
| A-222 | Redwood | 700 |
| A-217 | Brighton | 750 |

属性

- 一个关系的每个属性都有名字
 - 每个属性所允许的值的集合称为属性的域（ **domain** ）
 - 属性值要求必须是原子的，即不可再分
 - 如果域的所有成员是原子的，则说域是原子的
 - 特殊值 *null* 表示值未知或不存在
 - 空值在许多操作的定义中导致复杂性
 - 暂时忽略 **null** 值的副作用并在后面考虑它的作用
-

属性和关系有什么联系？

正常使用主观题需2.0以上版本雨课堂

作答

补充：对关系的形式化定义

- 属性 A_1, A_2, \dots, A_n 是抽象的符号
- D-Domain, 定义域, $D_i = \text{Dom}(A_i)$,
- **定义** 设 A_i 为属性, $D_i = \text{Dom}(A_i)$, $i=1, 2, \dots, n$.
- 则
 - 模式 $R(A_1, A_2, \dots, A_n)$ 上的关系 R 是 $D_1 \times D_2 \times \dots \times D_n$ 的一个子集合
 - R 的元素称为元组 tuple。
 - **等价说法:**
- a relation is a set of n-tuples (a_1, a_2, \dots, a_n)
- where, $a_i \in D_i$
-

□ Example: If

- ***customer_name*** = {Jones, Smith, Curry, Lindsay, ...} /* 所有客户名的集合 */
- ***customer_street*** = {Main, North, Park, ...} /* 所有街道名的集合*/
- ***customer_city*** = {Harrison, Rye, Pittsfield, ...} /* 所有城市名的集合 */

因此 $r = \{$
 (Jones, Main, Harrison),
 (Smith, North, Rye),
 (Curry, North, Rye),
 (Lindsay, Park, Pittsfield) $\}$

是一个关系, 它建立在(is a relation over)

customer_name \times *customer_street* \times *customer_city*

□ 注:

- **关系**是有限集 D_1, D_2, \dots, D_N 的**笛卡儿积的子集**，这个子集内的每个元组必须都是有实际意义的元组。
- 但是，笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 中的元组并不是每个都有实际意义，有些元组根本就没有实际意义。因此，从中取出有**实际意义**的元组来构成一个子集, 这个子集就是我们所要构造的关系。

属性

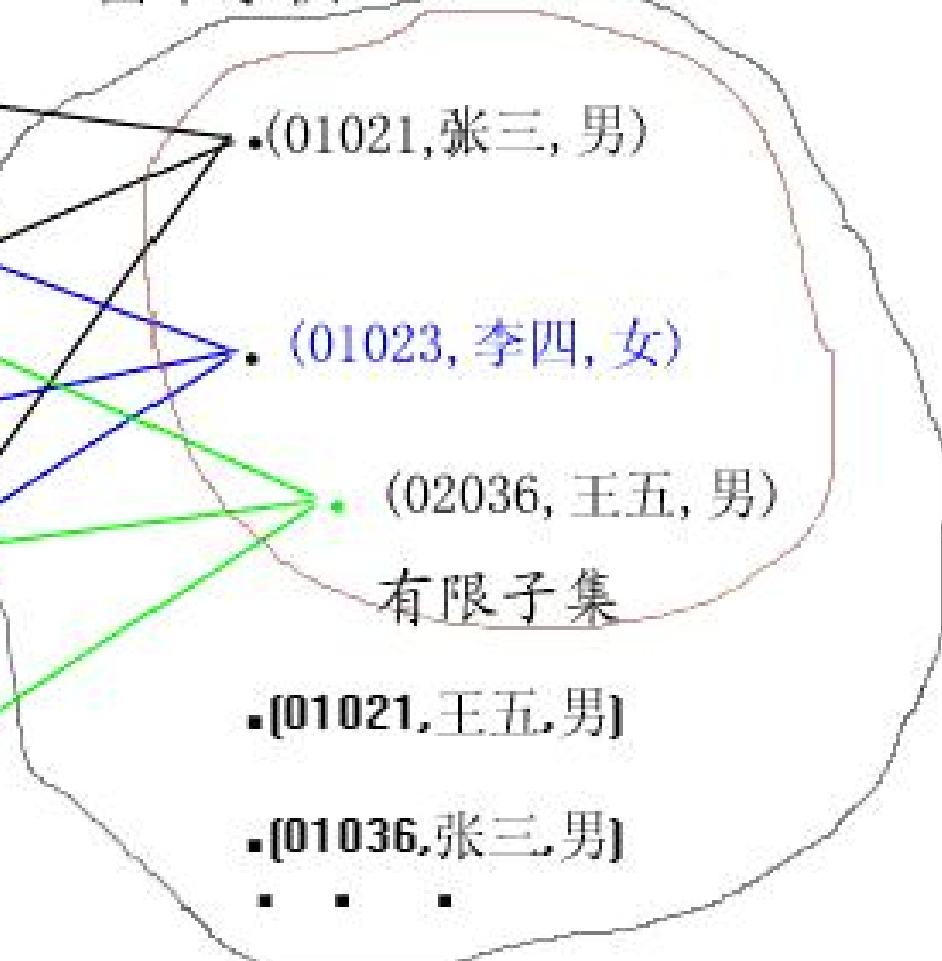
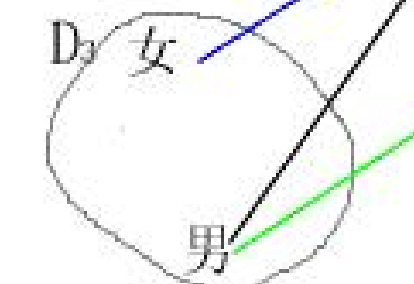
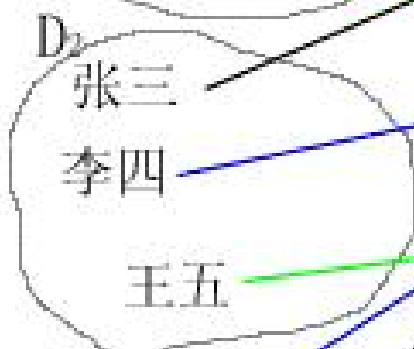
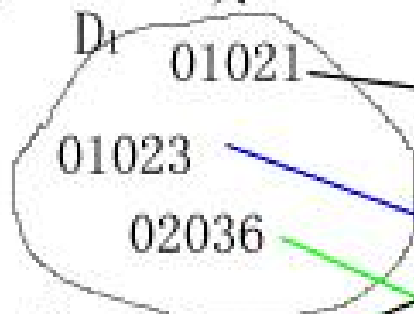
域

笛卡尔积 $D_1 \times D_2 \times D_3$

学号

姓名

性别



练习：若属性A1, A2分别在域D1, D2中取值（D1={1, 2, 3}，D2={a, b}）则，R1，R2，R3 是D1*D2上的关系吗？

$$D = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times \begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow D = \begin{bmatrix} 1 & a \\ 1 & b \\ 2 & a \\ 2 & b \\ 3 & a \\ 3 & b \end{bmatrix}$$

| R_1 | | R_2 | | R_3 | |
|-------|-------|-------|-------|-------|-------|
| A_1 | A_2 | A_2 | A_1 | A_2 | A_1 |
| 1 | a | 1 | a | a | 1 |
| 2 | b | 3 | b | b | 2 |
| 3 | a | 2 | a | a | 4 |

2.2 数据库模式

- 数据库模式

 - 数据库的逻辑设计

- 数据库实例

 - 在给定时刻数据库中数据的一个快照

课堂讨论：

分析关系模型的数据结构优缺点？

正常使用主观题需2.0以上版本雨课堂

作答

课堂讨论：阅读P29-30

教材中出现了几种码？

码有什么作用？

正常使用主观题需2.0以上版本雨课堂

作答

2.3 码 Keys

□ 理论上，要求：

- 必须有一种方法来区分一个关系中的不同元组
 - 一个关系中不能有两个元组在所有属性上取值完全相同
 - 码就是来区分关系中任意两条元组的
-

超码-superkey

- 设 $K \subseteq R$
 - K 是 R 的**超码**，如果 K 的值足以标识每个可能的关系 r (R) 的唯一的元组。
 - 例如： $\{ID\}$ and $\{ID, name\}$ 都是 *instructor* 的超码
-

候选码-candidate key

□ 如果 K 是最小的超码, 则 K 是候选码

如: $\{ID\}$ 是 *Instructor* 的候选码, 因为它是一个超码,
且它没有子集是超码;

□ 讨论: $\{ID, name\}$ 是候选码?

主码-Primary key

- 主码：在一个关系中被选为标识元组的候选码
 - 如： $\{ID\}$ 也是 *Instructor* 的主码，因为它是一个超码，且它没有子集是超码；
-

课堂实验：

在student表里增加相同学号的记录时，会产生什么问题？为什么？

正常使用主观题需2.0以上版本雨课堂

作答

Primary key的约束

- **实体完整性规则**：若 K 是基本关系 R 的主码，则构成 K 的各属性不能取空值，且任意两条元组在 K 上的取值不能相同。
- 讨论：一般情况下哪些属性可以作为主码？

属性本身取值就是唯一的
属性取值较少变化的

□ Example:

- 学生（学号，姓名，性别，班级）

学号为主码则其不能取空值。

在主码的属性下加下划线，并放在第一个位置。

外码-foreign key

□ Foreign Keys (外码)

- r_1 里包含的属性在 r_2 里做主码
- 对 r_1 来说, 这些属性叫外码, 参考 r_2
- r_1 : referencing relation 引用关系
- r_2 : referenced relation 被引用关系

r1的外码（一般情况下是单属性）

□ 在r1中:

- 外码一定不能是r1的主码

□ 在r2中:

- 为主码
- 或是候选码（在DBMS中，具有唯一性约束的属性）

□ 作用:

- 将两个表关联起来

□ 参照完整性约束:

- 保证两个表在关联属性上的取值一致性
-

□ 说明

- 关系 $r1$ 和 $r2$ 不一定是不同的关系;
 - 引用关系 $r1$ 的外码 F 和被引用关系 $r2$ 的主码 K_s 必须定义在同一个（或一组）域上;
 - 外码并不一定要与相应的主码同名;
 - 当外码与相应的主码属于不同关系时，往往取相同的名字，以便于识别;
-

课堂实验：

- ❑ 在sc表中增加学号为“811”同学的记录时，会产生什么问题？为什么？
 - ❑ 在student表中删除学号为“805”同学的记录时，会产生什么问题？为什么？
-

referential integrity constraint

参照完整性规则(对外码的取值约束)

- 若属性（或属性组） F 是关系 $r1$ 的外码，它与关系 $r2$ 的主码 K_s 相对应（关系 $r1$ 和 $r2$ 不一定是不同的关系），则对于 $r1$ 中每个元组在 F 上的值必须为：
- 或者取空值（ F 的每个属性值均为空值）；
 - 或者等于 $r2$ 中某个元组的主码值。
-

小组讨论：

- 找出下面关系模式中的外码、引用关系和被引用关系？

例1：学生实体、专业实体

学生（学号，姓名，性别，**专业号**，年龄）

专业（**专业号**，专业名）

学生（学号，姓名，性别，专业号，年龄）

| 学 号 | 姓 名 | 性 别 | 专 业 号 | 年 龄 |
|-------|-----|-----|-------|-----|
| 8 0 1 | 张 三 | 女 | 0 1 | 1 9 |
| 8 0 2 | 李 四 | 男 | 0 1 | 2 0 |
| 8 0 3 | 王 五 | 男 | 0 1 | 2 0 |
| 8 0 4 | 赵 六 | 女 | 0 2 | 2 0 |
| 8 0 5 | 钱 七 | 男 | 0 2 | 1 9 |

专业（专业号，专业名）

| 专 业 号 | 专 业 名 |
|-------|-------|
| 0 1 | 信 息 |
| 0 2 | 数 学 |
| 0 3 | 计 算 机 |

小组讨论:

讨论：下面关系模式中的外码能取空值吗？

例2 学生、课程、学生与课程之间的联系

学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

学生

| 学 号 | 姓 名 | 性 别 | 专 业 号 | 年 龄 |
|-------|-----|-----|-------|-----|
| 8 0 1 | 张 三 | 女 | 0 1 | 1 9 |
| 8 0 2 | 李 四 | 男 | 0 1 | 2 0 |
| 8 0 3 | 王 五 | 男 | 0 1 | 2 0 |
| 8 0 4 | 赵 六 | 女 | 0 2 | 2 0 |
| 8 0 5 | 钱 七 | 男 | 0 2 | 1 9 |

课程

| 课程号 | 课程名 | 学分 |
|-----|--------|----|
| 01 | 数据库 | 4 |
| 02 | 数据结构 | 4 |
| 03 | 编译 | 4 |
| 04 | PASCAL | 2 |

学生选课

| 学号 | 课程号 | 成绩 |
|-----|-----|----|
| 801 | 04 | 92 |
| 801 | 03 | 78 |
| 801 | 02 | 85 |
| 802 | 03 | 82 |
| 802 | 04 | 90 |
| 803 | 04 | 88 |

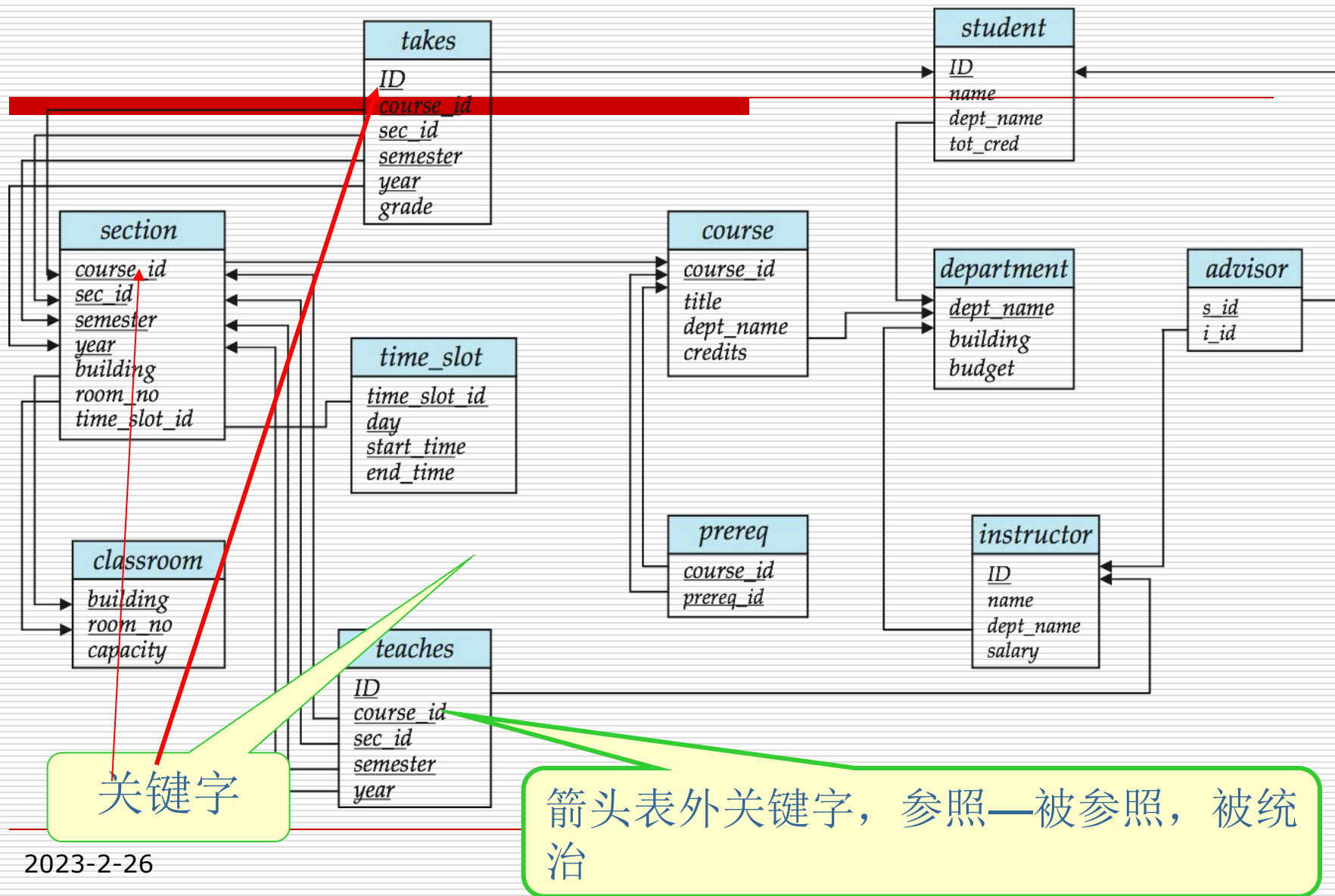
小组讨论： 下面关系模式中有外码吗？

例3 学生实体及其内部的领导联系

学生（学号，姓名，性别，专业号，年龄，班长）

| 学号 | 姓名 | 性别 | 专业号 | 年龄 | 班长 |
|-----|----|----|-----|----|-----|
| 801 | 张三 | 女 | 01 | 19 | 802 |
| 802 | 李四 | 男 | 01 | 20 | |
| 803 | 王五 | 男 | 01 | 20 | 802 |
| 804 | 赵六 | 女 | 02 | 20 | 805 |
| 805 | 钱七 | 男 | 02 | 19 | |

2.4 模式图



-
- 仔细分析P32的Fig 2-9，理解每个表包含的字段及表与表之间的关系。
 - 练习：
 - 请举出instructor表中的
 - Superkey
 - *candidate key*
 - Primary key
 - Foreign Keys
-

2.5 Query Languages

查询语言

- 它是用户用来从数据库中请求信息的语言.
- 语言的种类
 - 过程化的, Procedural
 - 非过程化的, Non-procedural, or declarative
- 纯语言 (“Pure” languages) :
 - 关系代数 (Relational algebra)
 - 元组关系演算 (Tuple relational calculus)
 - 域关系演算 (Domain relational calculus)
- 纯语言形成人们所使用的查询语言的底层基础.

Any Question ?