

上次课程主要内容

* 数据模型

- * 模型；数据模型；数据模型应满足的三个要求
 - * 概念模型、**逻辑模型**、物理模型
- * 组成要素：静态、动态、完整性约束
- * 概念模型 - ER图
- * 逻辑模型 - 关系模型

关系数据库

* 数据库系统结构

- * 模式、三级模式、两级映像
- * 逻辑独立性、物理独立性

关系数据库系统研究进展

- * 1970年 E.F.Codd – 系统、严格提出关系模型
“A Relational Model of Data for Shared Data Banks”
- * 20世纪70年代末 IBM公司在IBM370系列机研制关系数据库实验系统System R，历时六年。
- * 1981年 IBM公司，具有System R全部特征的数据库软件产品SQL/DS问世。
- * 同期，美国加州大学伯克利分校研制INGERS关系数据库实验系统，后由公司发展成INGERS。
- * 其他：ORACLE公司 ORACLE RDBMS、INFORMIX公司 INFORMIX、SYBASE公司 SYBASE、MicroSoft公司 SQL-SERVER、DBASE III→FoxPro、.....

关系数据库

孟 放

mengfang@cuc.edu.cn

3/25/2020

主要内容

- * 关系数据结构
- * 关系的完整性
- * 关系操作
 - * 关系代数
 - * 关系演算
- * 小结

主要内容

- * 关系数据结构

- * 关系的完整性

- * 关系操作

 - * 关系代数

 - * 关系演算

- * 小结

关系数据结构 – 基本概念

- * **关系**：逻辑结构为二维表 – DB中单一的数据结构。
- * **域**(Domain)：一组具有相同数据类型的值的集合。
- * **笛卡尔积**(Cartesian Product)：域上的一种集合操作。
 - * 给定一组域D1, D2, ..., Dn, （允许部分或全部相同）
- * D1, D2, ..., Dn的笛卡尔积为： **可表示成一个二维表**

$$D1 \times D2 \times \dots \times Dn = \{(d1, d2, \dots, dn) \mid di \in Dj, j = 1, 2, \dots, n\}$$

其中每一个元素(d1, d2, ..., dn)叫作一个n元组(n-Tuple),或简称为**元组**。元素中的每一个值di叫作一个**分量** (Component)。

若Di(i=1, 2, ..., n)为有限集, 其**基数**(Cardinal number)为mi(i=1, 2, ..., n), 则D1 × D2 × ... × Dn的基数为:

$$m = \prod_{i=1}^n m_i$$

关系数据结构 - 笛卡尔积

- * 设 $D1 = \text{男子集合(MAN)} = \{\text{王强、李东、张兵}\}$; $D2 = \text{女子集合(WOMAN)} = \{\text{赵红、吴芳}\}$; $D3 = \text{孩子集合(CHILD)} = \{\text{王一、李一、李二}\}$
- * 求上面三个集合的笛卡儿积
- * 构造一个家庭关系，可表示为：
FAMILY (MAN, WOMAN, CHILD)

Family

MAN	WOMAN	CHILD
王强	赵红	王一
李东	吴芳	李一
李东	吴芳	李二

M	W	C
王强	赵红	王一
王强	赵红	李一
王强	赵红	李二
王强	吴芳	王一
王强	吴芳	李一
王强	吴芳	李二
李东	赵红	王一
李东	赵红	李一
李东	赵红	李二
李东	吴芳	王一
李东	吴芳	李一
李东	吴芳	李二
张兵	赵红	王一
张兵	赵红	李一
张兵	赵红	李二
张兵	吴芳	王一
张兵	吴芳	李一
张兵	吴芳	李二

关系数据结构 – 基本概念

- * 关系 (Relation)

- * $D_1 \times D_2 \times \cdots \times D_n$ 的有意义的子集称为在域 D_1, D_2, \cdots, D_n 上的关系, 记为 $R(D_1, D_2, \cdots, D_n)$ 。
- * 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, 是另一关系的码, 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 - 例

- * 学生: S (Sno, SNAME, SA, SD)
- * 课程: C (Cno, CNAME, PC#)
- * 选课: SC (Sno, Cno, GR)
- * S: 候选码: Sno, SNAME; 主码: Sno
- * C: 候选码: Cno; 主码: Cno
- * SC: 候选码: (Sno, Cno); 主码: (Sno, Cno); 外部码: Sno, Cno

* 关系性质

1. 每一列中的数值是同类型的数据, 来自同一个域。
2. 不同的列可对应于同一个域, 但需给予不同的属性名。
3. 同一关系中任意两个元组的候选码不能相同。无重复行。
4. 行(列)的次序可以任意交换, 不影响关系的实际意义。
5. 关系中的每一个属性值都必须是不能再分的元素。

关系数据结构 – 关系模式

* 引入

- * 关系模式是**型**。是静态的、稳定的；
- * 关系是关系模式的**值**。是动态的、随时间而变化的。

* 定义：对关系的描述称为关系模式，记为 $R(U, D, DOM, F)$ 。R为关系名。

- * U为属性集
- * D为U所对应的域的集合
- * DOM为属性向域的映象集合
- * F为属性间数据依赖关系的集合

* 关系模式通常简记为: $R(U)$

- * $S(Sno, Sname, Sage, Sdept)$

针对一个具体问题，如何构造适合它的关系模式？

* 学生选课系统 – 定义的关系模式

SCS(Sno, Sname, Sage, Sdept, Cno, Cname, Cpname, Grade, Deputy)

◆ 存在问题：

- ◆ 数据冗余，更新（增/删/改）异常，……

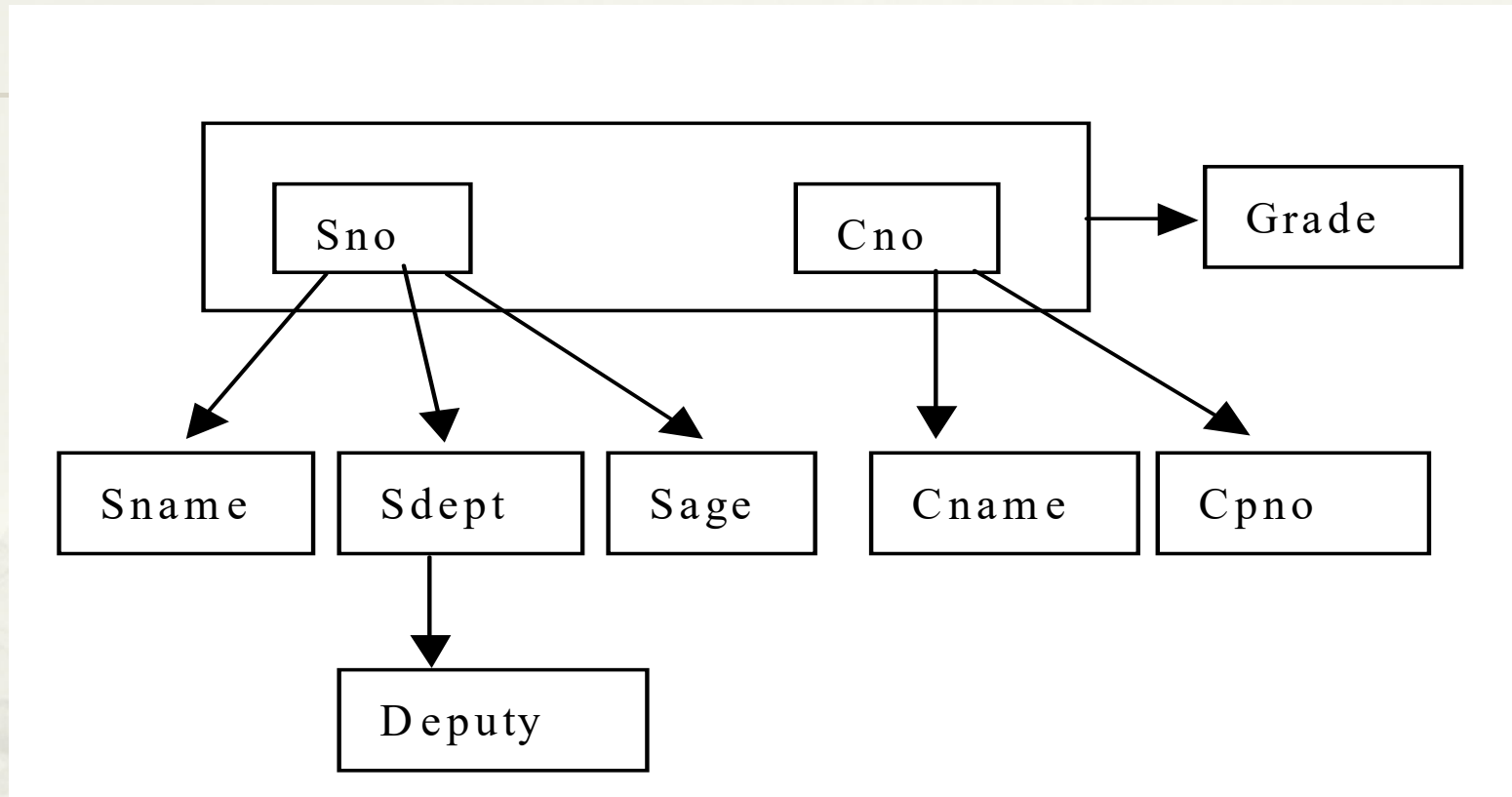
◆ 原因：

- ◆ 包罗万象，内容太杂！
- ◆ 存在“不恰当”的依赖关系。

关系数据理论 - 基础知识

- * 数据依赖
 - * 一个关系内部属性与属性之间的一种约束关系。
- * 范式NF(Normal Form)
 - * 关系数据库中的关系是要满足一定要求的；满足不同程度要求则成为不同范式。
- * 1NF：第一范式
 - * 关系中每一个分量不可再分的范式。【最低要求】
- * 规范化normalization
 - * 一个低一级范式的关系模式通过模式分解转化为若干个高一级范式的关系模式的集合。
- * 关系模式的简化表示： $R(U, F)$

选课关系所定义模式的规范化



- * $F = \{ Sno \rightarrow Sname, Sno \rightarrow Sdept, Sno \rightarrow Sage, Sdept \rightarrow Deputy, Cno \rightarrow Cname, Cno \rightarrow Cpno, (Sno, Cno) \rightarrow Grade \}$

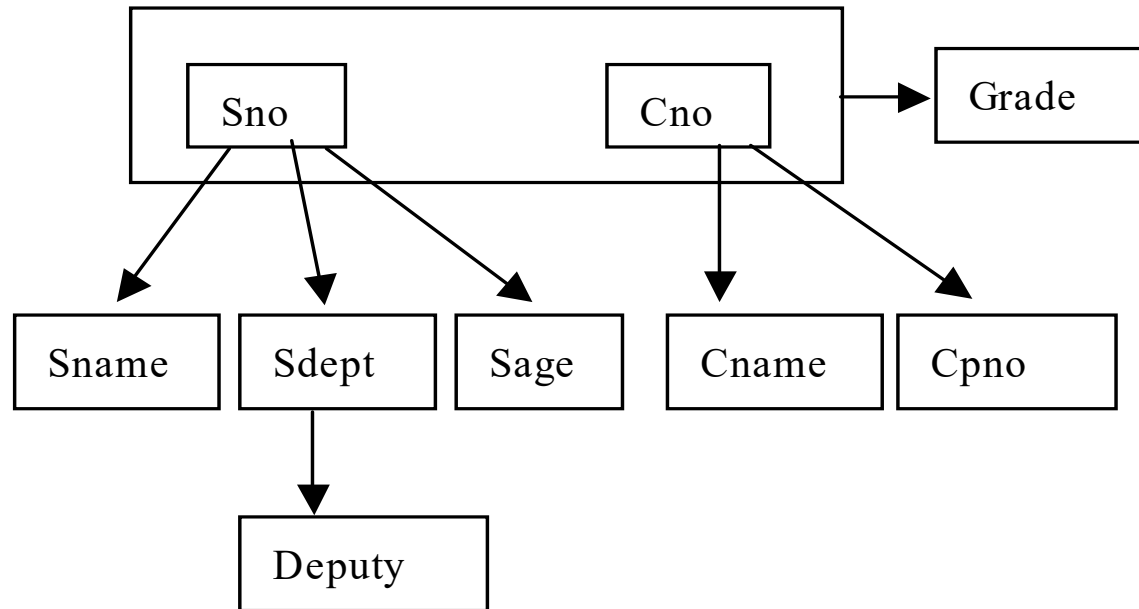
模式分解 - 关系模式的规范化

* 将原来的关系模式

SCS(Sno, Sname, Sage, Sdept, Cno, Cname, Cpno, Grade, Deputy)

* 分解如下：

- **S(Sno, Sname, Sdept, Sage)**
- **C(Cno, Cname, Cpno)**
- **D(Sdept, Deputy)**
- **SC(Sno, Cno, Grade)**



范式及规范化小结

- * 规范化**目的**：为解决仅满足第一范式（基本要求）的关系模式所存在插入、删除异常、修改复杂和数据冗余等问题。
- * 规范化**基本思想**：逐步消除数据依赖中不合适的地方，追求**概念的单一化**。
- * 规范化**过程**：
 - * 2NF：消除非主属性对码的部分函数依赖
 - * 3NF：消除非主属性对码的传递函数依赖
 - * BCNF：消除主属性对码的部分和传递函数依赖
 - * 4NF：消除非平凡且非函数依赖的多值依赖

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。

主要内容

- * 关系数据结构
- * 关系的完整性
- * 关系操作
 - * 关系代数
 - * 关系演算
- * 小结

关系的完整性

- * 定义：对关系的某种约束性。
 - * 关系的值随时间变化是应该满足的一些约束条件 - **现实世界的要求**。
- * 三类完整性约束
 - * 关系模型必须满足的完整性约束条件
 - * 实体完整性
 - * 参照完整性
 - * 应用系统需要遵循的约束条件
 - * 用户自定义完整性

关系完整性 - 实体完整性

- * 实体完整性 Entity Integrity
 - * 若A是关系R(U) ($A \in U$) 上的主属性，则属性A不能取空值。
 - * 如，选课SC中的Sno, Cno均不能取空值。
- * 说明
 - * 实体完整性是针对基本关系
 - * 现实世界的实体是可区分的，即，实体具有某种唯一性标识
 - * 而关系模型以主码作为实体的唯一性标识
 - * 故，主码中的属性不能取空值

关系完整性 - 参照完整性

* 参照完整性 Referential Integrity

- * 属性（属性组） X 是关系 R 的外部码， K_s 是关系 S 的主码，且 X 与 K_s 相对应（即 X, K_s 是定义在同一个域上），则 R 中任一元组在 X 上的值为： $X = \text{空值或} S \text{中的某个元组的} K_s \text{值。}$

- * 如，职工 EMP(EMP#, ENAME, JOB, DEPT#)

部门 DEPT(DEPT#, DNAME, LOC)

则：EMP中的DEPT#为空 或 为DEPT中的DEPT#的值

* 定义外码与主码之间的引用规则

关系完整性 - 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。
- * 完整性约束由数据库管理系统DBMS保证

主要内容

- * 关系数据结构
- * 关系的完整性
- * 关系操作
 - * 关系代数
 - * 关系演算
- * 小结

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
 - * 选择Select、投影Project、连接Join、除Divide、并Union、差Except、交Intersection、笛卡儿积、等

关系操作 – 关系数据语言分类

- * 关系代数
 - * 用关系运算来表达查询，以ISBL为代表
- * 关系演算：用谓词来表达查询
 - * 元组演算：用谓词公式来表达查询元组演算(以行为变量)，以ALPHA为代表
 - * 域演算：域演算(以列为变量)，以QBE为代表
- * 关系数据库标准语言SQL
 - * 具有关系代数和关系演算双重特点

关系代数与关系演算

* 关系代数与关系演算

- * 关系代数语言：查询操作是以**集合操作**作为基础的语言
- * 关系演算语言：查询操作是以**谓词演算**作为基础的语言

□ 关系数据语言是一种比Pascal、C等程序设计语言更高级的语言。

□ Pascal、C一类语言属于过程性语言，在编程时必须给出获得结果的操作步骤。

□ 而关系数据语言属于非过程性语言，编程时只需要指出需要什么信息，不必给出具体的操作步骤。

干什么？
怎么干？

干什么？

主要内容

- * 关系数据结构
- * 关系的完整性
- * 关系操作
 - * 关系代数
 - * 关系演算
- * 小结

关系代数 – 相关的表述符号

- * 设关系模式为 $R(A_1, A_2, \dots, A_n)$ 。它的一个关系设为 R 。 $t \in R$ 表示 t 是 R 的一个元组。 $t[A_i]$ 表示元组 t 中对应属性 A_i 的一个分量。
- * 若 $A = \{A_{i1}, A_{i2}, \dots, A_{ik}\}$ ，其中 $A_{i1}, A_{i2}, \dots, A_{ik}$ 是 A_1, A_2, \dots, A_n 中的一部分，则 A 称为属性列或域列。 \bar{A} 则表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 $\{A_{i1}, A_{i2}, \dots, A_{ik}\}$ 后剩余的属性组。 $t[A] = (t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}])$ 表示元组 t 在属性列 A 上诸分量的集合。
- * R 为 n 目关系， S 为 m 目关系。 $r \in R, s \in S$ 。 rs 称为元组的连接。它是一个 $(n+m)$ 列的元组，前 n 个分量为 R 中的一个 n 元组，后 m 个分量为 S 中的一个 m 元组。
- * 给定一个关系 $R(X, Z)$ ， X 和 Z 为属性组。我们定义，当 $t[X] = x$ 时， x 在 R 中的象集 (Images Set) 为： $Zx = \{ t[Z] \mid t \in R, t[X] = x \}$ 它表示 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合。

关系代数 – 传统集合运算

- 并运算
- 差运算
- 交运算
- 笛卡尔积

R1

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

R2

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

$R1 \cup R2$

A	B	C
a1	b1	c1
a1	b2	c2
a1	b3	c2
a2	b2	c1

$R1 - R2$

A	B	C
a1	b1	c1

$R1 \cap R2$

A	B	C
a1	b2	c2
a2	b2	c1

关系代数 - 专门的关系运算

- * 以学生关系S为例
- * 关系运算
 - * 选择
 - * 投影
 - * 连接
 - * 除

学号	姓名	系名	年龄
Sno	SN	SD	SA
S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

关系运算 - 选择运算

学号	姓名	系名	年龄
Sno	SN	SD	SA
S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

- * 选择Selection: 行的角度。
 - * 在关系R中选择满足给定条件的元组, 记做:

$$\sigma_F(R) = \{ t \mid t \in R \wedge F(t) = \text{'真'} \}$$
 - * F是一个公式, 表示形式为由逻辑运算符 (\wedge, \vee, \neg) 连接各算术表达式组成。
 - * 算术表达式的基本形式为: $X\theta Y$. $\theta = \{>, \geq, <, \leq, =, \neq\}$. X,Y是属性名或常量或简单函数。

如: 查询计算机系(CS)学生,

即, $\sigma_{SD='CS'}(S)$

(S')

$$\sigma_{SD='CS'}(S)$$

Sno	SN	SD	SA
S1	A	CS	20
S2	B	CS	21
S6	F	CS	22

关系运算 - 投影运算

学号 Sno	姓名 SN	系名 SD	年龄 SA
S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

* 投影Projection: 列的角度

- * 关系R上的投影是从R中选择若干属性组A成新的关系。记做:

$$\pi_A(R) = \{ t[A] \mid t \in R \}$$

- * 投影之后不仅取消了某些列, 还可能取消某些元组。

- * 如: 求解学生关系S在学生姓名和所在系这两个属性上的投影结果。

即, $\pi_{SN,SD}(S)$

$\pi_{SN,SD}(S)$

SN	SD
A	CS
B	CS
C	MA
D	CI
E	MA
F	CS

$\pi_{SA}(S)$

SA
20
21
19
22

关系运算 - 连接运算

- * 连接Join: 从两个关系的笛卡尔积中选取属性间满足一定条件的元组。

记作: $R \underset{F}{\bowtie} S$.

其中, F是条件表达式, 它涉及到对两个关系中的属性的比较。如果F仅仅是一个相等条件, 即等值连接。无F时为自然连接。

$R \bowtie S$

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2

- * 例

R

A	B	C
a1	b1	5
a1	b2	6
a2	b3	8
a2	b4	12

S

B	E
b1	3
b2	7
b3	10
b3	2
b5	2

$R \underset{C < E}{\bowtie} S$

A	R.B	C	S.B	E
a1	b1	5	b2	7
a1	b1	5	b3	10
a1	b2	6	b2	7
a1	b2	6	b3	10
a2	b3	8	b3	10

关系运算 - 除运算

* 除运算Division

- * 给定关系 $R(X,Y)$ 和 $S(Y,Z)$ ，其中 X, Y, Z 为属性组。 R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须出自相同的域集。 R 与 S 的除运算得到一个新的关系 $P(X)$ ， P 是 R 中满足下列条件的元组在 X 属性列上的投影：元组在 X 上分量值 x 的象集 Y_x 包含 S 在 Y 上投影的集合。记作：

$$R \div S = \{ t[X] \mid t \in R \wedge Y_x \supseteq \Pi_Y(S) \}$$

- * 其中 Y_x 为 x 在 R 中的象集， $x=r[X]$ 。

关系运算 – 除运算 – 例

* 设关系R、S如下，求解 $R \div S$

* a_1 的象集 $\{(b_1, c_2), (b_2, c_3), (b_2, c_1)\}$

* a_2 的象集 $\{(b_3, c_7), (b_2, c_3)\}$

* a_3 的象集 $\{(b_4, c_6)\}$

* a_4 的象集 $\{(b_6, c_6)\}$

* S在(B,C)属性组的投影，为

$\{(b_1, c_2), (b_2, c_1), (b_2, c_3)\}$

只有 a_1 的象集包含了S在(B,C)属性组的投影，

故 $R \div S = \{a_1\}$

A	B	C
a_1	b_1	c_2
a_2	b_3	c_7
a_3	b_4	c_6
a_1	b_2	c_3
a_4	b_6	c_6
a_2	b_2	c_3
a_1	b_2	c_1

B	C	D
b_1	c_2	d_1
b_2	c_1	d_1
b_2	c_3	d_2

关系运算 - 查询 - 练习题目

- * 已知学生课程数据库中包含学生关系S，课程关系C和选修关系SC，关系模式分别为
 - * S(Sno, Sname, Sgender, Sage, Sdept)
 - * C(Cno, Cname, Cpno, Ccredit)
 - * SC(Sno, Cno, Grade)
- * 试写出如下查询要求的代数表达式
 - * 查询计算机系的全体学生；
 - * 查询都有哪些系；
 - * 查询至少选修1号课程和3号课程的学生学号；
 - * 查询选修了2号课程的学生学号；
 - * 查询至少选修了一门其先修课程为5号课程的学生姓名；
 - * 查询选修了全部课程的学生学号和姓名。

小结

- * 关系数据结构
- * 关系的完整性
- * 关系操作
 - * 关系代数
 - * 关系演算（自学）

作业

- * 课堂练习题目，书上有答案。
- * P.70: 3、5、6(只用“关系代数”求解)