数据库系统概论

Introduction to Database Systems

第2章　关系模型

中国人民大学信息学院

关系模型

- 提出关系模型的是美国IBM公司的E.F.Codd
  - 1970年提出关系数据模型

    E.F.Codd, “A relational Model of Data for Large Shared Data Banks”, 《Communications of the ACM》,1970
  - 20世纪70年代末，关系方法理论和软件系统研制紧密结合，取得丰硕成果
    - IBM公司的San Jose实验室研制的System R获得成功
    - UCBerkley 研制了INGRES关系数据库实验系统，1980年代中期又研发了PostgresSQL系统，并成功开放源代码
    - 1978年Oracle公司成立发布了Oracle1.0，并不断发展成熟

第2章 关系数据库

2.1 关系模型的数据结构及形式化定义

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构

2.1.1 关系

- 单一的数据结构----关系

  现实世界的实体以及实体间的各种联系均用关系来表示

- 逻辑结构----二维表

  从用户角度，关系模型中数据的逻辑结构是一张二维表

- 建立在集合代数的基础上

关系（续）

1. 域（domain）
2. 笛卡儿积（Cartesian product）
3. 关系（relation）

1. 域（domain）

- 域是一组具有相同数据类型的值的集合。
- 例:
  - 整数
  - 实数
  - 介于某个取值范围的整数
  - 长度小于25B的变长字符串集合
  - {男，女}
  - ……

2. 笛卡儿积（Cartesian product）

❖笛卡儿积

给定一组域D_1，D_2，…，D_n，允许其中某些域是相同的。

D_1，D_2，…，D_n的笛卡儿积为：

$D_1 \times D_2 \times \ldots \times D_n =$

$\{（d_1，d_2，\ldots，d_n）\mid d_i \in D_i，i=1，2，\ldots，n\}$

■所有域的所有取值的一个组合

■不能重复

笛卡儿积（续）

- **元组（tuple）**
  - 笛卡儿积中每一个元素（d_1，d_2，…，d_n）叫作一个n元组（n-tuple）或简称元组
  - (张清玫，计算机科学与技术，李勇)、
  - (张清玫，计算机科学与技术，刘晨) 等都是元组
- **分量（Component）**
  - 笛卡儿积元素（d_1，d_2，…，d_n）中的每一个值d_i叫做一个**分量**
  - 张清玫、计算机科学与技术、李勇、刘晨等都是分量

笛卡儿积（续）

- 基数（cardinal number）
  - 一个域允许的不同取值个数
  - 若D_i（i=1，2，…，n）为有限集，其基数为m_i（i=1，2，…，n），则$D_1 \times D_2 \times \ldots \times D_n$的基数$M$为：

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

- 笛卡儿积的表示方法
  - 可表示为一张二维表
  - 表中的每行对应一个元组，表中的每列来自一个域

笛卡儿积（续）

例如，给出3个域：

- D_1=导师集合SUPERVISOR=｛张清玫，刘逸｝
- D_2=专业集合MAJOR=｛计算机科学与技术，信息管理与信息系统｝
- D_3=研究生集合POSTGRADUATE=｛李勇，刘晨，王敏｝
- D_1，D_2，D_3的笛卡儿积为

笛卡儿积（续）

- D1×D2×D3＝｛
  (张清玫，计算机科学与技术，李勇)，(张清玫，计算机科学与技术，刘晨)，
  (张清玫，计算机科学与技术，王敏)，(张清玫，信息管理与信息系统，李勇)，
  (张清玫，信息管理与信息系统，刘晨)，(张清玫，信息管理与信息系统，王敏)，
  (刘逸，计算机科学与技术，李勇)，(刘逸，计算机科学与技术，刘晨)，
  (刘逸，计算机科学与技术，王敏)，(刘逸，信息管理与信息系统，李勇)，
  (刘逸，信息管理与信息系统，刘晨)，(刘逸，信息管理与信息系统，王敏)｝
- 基数为2×2×3＝12

笛卡儿积（续）

表2.1 笛卡儿积示例

SUPERVISOR 导师	MAJOR 专业	POSTGRADUATE 研究生
张清玫	计算机科学与技术	李勇
张清玫	计算机科学与技术	刘晨
张清玫	计算机科学与技术	王敏
张清玫	信息管理与信息系统	李勇
张清玫	信息管理与信息系统	刘晨
张清玫	信息管理与信息系统	王敏
刘逸	计算机科学与技术	李勇
刘逸	计算机科学与技术	刘晨
刘逸	计算机科学与技术	王敏
刘逸	信息管理与信息系统	李勇
刘逸	信息管理与信息系统	刘晨
刘逸	信息管理与信息系统	王敏

3. 关系（relation）

- 关系模型中D_1，D_2，…，D_n的笛卡儿积一般没有实际语义，只有某个真子集才有实际含义
- 表2.1的笛卡儿积中许多元组是没有意义的
  - 在学校中一个专业方向有多个导师，而一个导师只在一个专业方向带研究生；
  - 一个导师可以带多名研究生，而一名研究生只有一个导师，学习某一个专业。
  - 表2.1中的一个子集才是有意义的，才可以表示导师与研究生的关系，把该关系取名为SMP

关系（续）

- 把关系SMP属性名取为SUPERVISOR，MAJOR和POSTGRADUATE
- 导师-研究生关系模式可以表示为SMP（SUPERVISOR，MAJOR，POSTGRADUATE）

表2.2 导师-研究生关系SMP

SUPERVISOR	MAJOR	POSTGRADUATE
张清玫	计算机科学与技术	李勇
张清玫	计算机科学与技术	刘晨
刘逸	信息管理与信息系统	王敏

关系（续）

（1） 关系

$D_1 \times D_2 \times \ldots \times D_n$的子集叫作在域$D_1$，$D_2$，…，$D_n$上的关系，表示为

$R(D_1, D_2, \ldots, D_n)$

- R：关系名
- n：关系的目或度（degree）

关系（续）

（2）元组

关系中的每个元素是关系中的元组，通常用*t*表示

（3）单元关系与二元关系

当*n*=1时，称该关系为**单元关系（unary relation）**

或一元关系

当*n*=2时，称该关系为**二元关系（binary relation）**

关系（续）

（4）关系的表示

关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域

（5）属性

- 关系中不同列可以对应相同的域
- 为了加以区分，必须对每列起一个名字，称为属性（attribute）
- *n*目关系必有*n*个属性
  - ✓ SMP（SUPERVISOR，MAJOR，POSTGRADUATE）有3个属性，是一个3目关系
  - ✓ （张清玫，计算机科学与技术，李勇），（张清玫，信息管理与信息系统，刘晨），(刘逸，信息管理与信息系统，王敏) 是SMP关系的3个元组

关系（续）

（6）三类关系

基本关系（基本表或基表）

实际存在的表，是实际存储数据的逻辑表示

查询结果

查询执行产生的结果对应的临时表

视图表

由基本表或其他视图表导出的虚表，不存储实际数据

关系（续）

（7）基本关系的性质

① 列是同质的（homogeneous）

② 不同的列可出自同一个域

- 其中的每一列称为一个属性
  - 不同的属性要给予不同的属性名

③ 列的顺序无所谓,列的次序可以任意交换

④ 任意两个元组的码不能相同

⑤ 行的顺序无所谓，行的次序可以任意交换

基本关系的性质（续）

⑥ 分量必须取原子值，即每一个分量都必须是不可分的数据项
这是规范条件中最基本的一条

表2.3 非规范化关系

SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	计算机科学与技术	李勇	刘晨
刘逸	信息管理与信息系统	王敏	

小表

2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构

2.1.2 关系模式

1. 什么是关系模式
2. 定义关系模式
3. 关系模式与关系

1. 什么是关系模式

- 关系模式是型，关系是值
- 关系模式是对关系的描述
  - 描述关系元组集合的结构
    - 属性构成
    - 属性来自的域
    - 属性与域之间的映象关系
  - 描述关系的完整性约束

2. 定义关系模式

关系模式可以形式化地表示为：

R（U，D，DOM，F）

R	关系名
U	组成该关系的属性名集合
D	***U***中属性所来自的域
DOM	属性向域的映象集合
F	属性间数据的依赖关系集合

定义关系模式（续）

例:

导师和研究生出自同一个域—人，

■取不同的属性名

■在模式中定义属性向域的映象，即说明它们分别出自哪个域

DOM（SUPERVISOR）
= DOM（POSTGRADUATE）
= PERSON

定义关系模式（续）

关系模式通常可以简记为

$R(U)$ 或 $R(A_1, A_2, \ldots, A_n)$

- R: 关系名
- $A_1, A_2, \ldots, A_n$：属性名
- 域名及属性向域的映象常常直接说明为属性的类型、长度

定义关系模式（续）

- 候选码（candidate key）

  关系模式中的某一个属性或一组属性的值能唯一地标识一个元组，而它的真子集不能唯一地标识一个元组，则称该属性或属性组为候选码为候选码

  简单的情况：候选码只包含一个属性

- 全码（all-key）

  最极端的情况：关系模式的所有属性是这个关系模式的候选码，称为全码（all-key）

定义关系模式（续）

- 主码

若一个关系有多个候选码，则选定其中一个为主码（primary key）

例如：在导师-研究生关系SMP（SUPERVISOR，MAJOR，POSTGRADUATE）中，假设研究生不会重名，则POSTGRADUATE可以作为SMP关系的主码，用下划线表示

- 主属性

候选码的诸属性称为主属性（prime attribute）

不包含在任何侯选码中的属性称为非主属性（non-prime attribute）或非码属性（non-key attribute）

3. 关系模式与关系

- 关系模式
  - 对关系的描述
  - 静态的、稳定的
- 关系
  - 关系模式在某一时刻的状态或内容
  - 动态的、随时间不断变化的
- 关系模式和关系往往笼统称为关系

  通过上下文加以区别

2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构

2.1.3 关系数据库

- 关系数据库系统
  - 支持关系模型的数据库系统
  - 关系模型中，实体以及实体间的联系都用关系表示例如学生实体、课程实体、学生与课程之间选修课程的多对多联系
  - 在一个关系数据库中，某一时刻所有关系模式对应的关系的集合构成一个关系数据库

关系数据库（续）

■学生关系模式：Student(Sno, Sname, Ssex, Sbirthdate, Smajor)

包括学号、姓名、性别、出生日期和主修专业等属性

学号 Sno	姓名 Sname	性别 Ssex	出生日期 Sbirthdate	主修专业 Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	刘晨	女	1999-9-1	计算机科学与技术
20180003	王敏	女	2001-8-1	计算机科学与技术
20180004	张立	男	2000-1-8	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180006	赵明	男	2000-6-12	数据科学与大数据技术
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

关系数据库（续）

- 课程关系模式：Course(Cno,Cname,Ccredit,Cpno)

  包括课程号，课程名，学分，先修课（直接先修课）等

课程号 Cno	课程名 Cname	学分 Ccredit	先修课 Cpno
81001	程序设计基础与C语言	4	
81002	数据结构	4	81001
81003	数据库系统概论	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

关系数据库（续）

- 学生选课关系模式：SC(Sno,Cno, Grade,Semester,Teachingclass)

  包括学号，课程号，成绩，选课学期，教学班等

学号 Sno	课程号 Cno	成绩 Grade	选课学期 Semester	教学班 Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

关系数据库（续）

- 关系数据库也有型和值之分
- 关系数据库的型
  - 关系数据库中所有关系模式的集合
  - 是对关系数据库的描述
  - 通常称为关系数据库模式
- 关系数据库的值
  - 这些关系模式在某一时刻对应的关系的集合
  - 通常称为关系数据库

2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.4 关系模型的存储结构

2.1.4 关系模型的存储结构

- 关系数据库管理系统以一定的组织方式来存储和管理数据，即设计和实现关系模型的存储结构。
- 有的关系数据库管理系统中一个表对应一个操作系统文件，将物理数据组织的任务交给操作系统完成
- 有的关系数据库管理系统从操作系统那里申请若干个大的文件，自己划分文件空间，组织表、索引等存储结构，并进行存储管理

第2章 关系数据库

2.2.1 基本的关系操作

- 常用的关系操作
  - ■查询操作：选择、投影、连接、除、并、差、交、笛卡儿积
    - ●选择、投影、并、差、笛卡儿积是5种基本操作
  - ■更新操作：插入、删除、修改
- 关系操作的特点
  - ■集合操作方式：操作的对象和结果都是集合，一次一个集合的方式
  - ■关系操作的所有输入和输出均是关系，包括关系操作的中间结果也是关系

2.2.2 关系数据语言的分类

- 关系代数语言
  - 用对关系的运算来表达查询要求
  - 代表：ISBL
- 关系演算语言：用谓词来表达查询要求
  - 元组关系演算语言
    - 谓词变元的基本对象是元组变量
    - 代表：APLHA, QUEL
  - 域关系演算语言
    - 谓词变元的基本对象是域变量
    - 代表：QBE
  - 结构化查询语言（Structured Query Language， SQL ）
    - 具有关系代数和关系演算双重特点

第2章 关系数据库

关系的完整性

- 关系的三类完整性约束
  - 实体完整性和参照完整性
    - 关系模型必须满足的完整性约束条件称为关系的两个不变性，应该由关系系统自动支持
  - 用户定义的完整性
    - 应用领域需要遵循的约束条件，体现了具体领域中的语义约束

2.3 关系的完整性

2.3.1 实体完整性

2.3.2 参照完整性

2.3.3 用户定义的完整性

2.3.1 实体完整性

❖ 规则2.1 实体完整性规则（entity integrity）

■若属性（指一个或一组属性）*A*是基本关系*R*的主属性，则*A*不能取空值

■空值就是“不知道”或“不存在”或“无意义”的值

例：

学生选课（学号，课程号，成绩，选课学期，教学班）

（学号、课程号）为主码

“学号”和“课程号”两个属性都不能取空值

实体完整性（续）

❖实体完整性规则的说明

（1）实体完整性规则是针对基本关系而言的

一个基本表通常对应现实世界的一个实体集

（2）现实世界中的实体是可区分的，即它们具有某种唯一性标识

（3）关系模型中以主码作为唯一性标识

（4）主码中的属性不能取空值

如果取了空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与第（2）点相矛盾，因此这个规则称为实体完整性

2.3 关系的完整性

2.3.1 实体完整性

2.3.2 参照完整性

2.3.3 用户定义的完整性

2.3.2 参照完整性

1. 关系间的引用
2. 外码
3. 参照完整性约束

1. 关系间的引用

❖在关系模型中实体及实体间的联系都是用关系来描述的，自然存在着关系与关系间的引用

[例2.1] “学生”实体、“专业”实体

主码

学生（学号，姓名，性别，出生日期，主修专业）

专业（专业名，专业编号）

主码

- 学生关系引用了专业关系的主码“专业名”。
- 学生关系中的“主修专业”值必须是确实存在的专业名

关系间的引用（续）

例[2.2] 学生、课程、学生与课程之间的多对多联系

学生（学号，姓名，性别，出生日期，主修专业）

课程（课程号，课程名，学分，先修课）

学生选课（学号，课程号，成绩，选课学期，教学班）

- 学生选课关系引用学生关系的主码“学号”和课程关系的主码“课程号”
- 学生选课关系中的“学号”值必须是确实存在的学生的学号
- 学生选课关系中的“课程号”值也必须是确实存在的课程的课程号

关系间的引用（续）

❖同一关系内部属性间也可能存在引用关系

例[2.3]在课程（课程号，课程名，学分，先修课）中

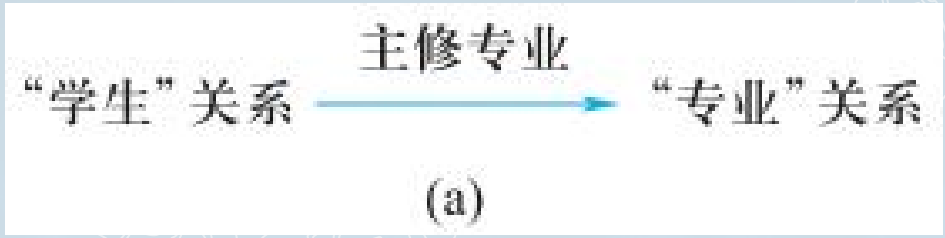
- “课程号”属性是主码
- “先修课”属性表示选修该门课程之前需要完成先修课程的课程号，它引用了本关系“课程号”属性，即“课程号”必须是确实存在的课程的课程号

2．外码（foreign key）

- 设*F*是基本关系*R*的一个或一组属性，但不是关系*R*的码，Ks是基本关系S的主码。如果F与Ks相对应，则称F是*R*的外码
- 基本关系*R*称为参照关系（referencing relation）
- 基本关系*S*称为被参照关系（referenced relation）或目标关系（target relation）

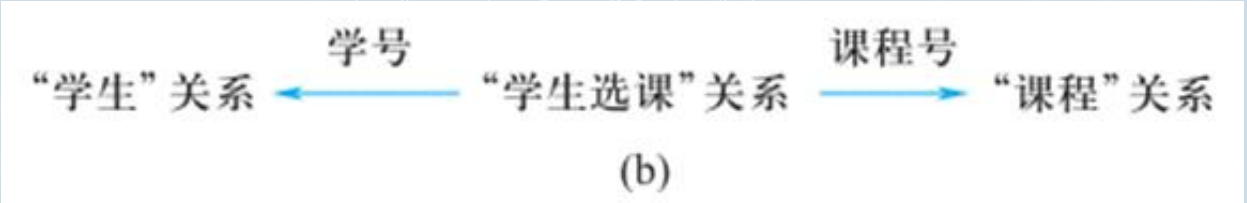
外码（续）

- 例2.1“学生”关系的“主修专业”属性与“专业”关系的主码“专业名”相对应
  - “主修专业”属性是学生关系的外码
  - “专业”关系是被参照关系，“学生”关系为参照关系



外码（续）

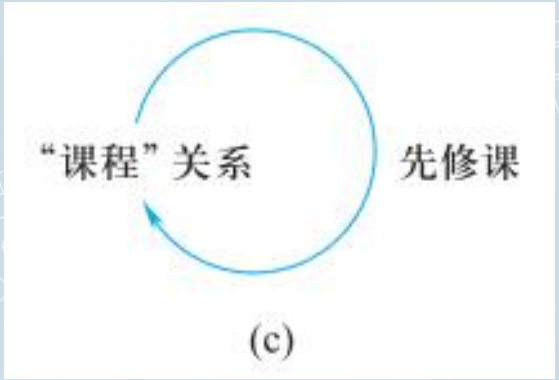
- [例2.2]中
  - “学生选课”关系的“学号” 与“学生”关系的主码“学号”相对应
  - “ 学生选课”关系的“课程号”与“课程”关系的主码“课程号”相对应
  - “学号”和“课程号”是“学生选课”关系的外码
  - “学生”关系和“课程”关系均为被参照关系
  - “学生选课”关系为参照关系



(b)

外码（续）

- [例2.3] “课程”关系中“先修课”属性与本身的主码“课程号”属性相对应
  - “先修课”是外码
  - “课程”关系既是参照关系也是被参照关系



外码（续）

- 关系*R*和*S*不一定是不同的关系
- 目标关系*S*的主码Ks 和参照关系的外码F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

3. 参照完整性约束

❖ 规则2.2 参照完整性约束

若属性（或属性组）*F*是基本关系*R*的外码，它与基本关系*S*的主码Ks相对应（基本关系*R*和*S*不一定是不同的关系），则对于*R*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*S*中某个元组的主码值

参照完整性约束（续）

[例2.1]中

“学生”关系中每个元组的“主修专业”属性只取两类值：

（1）空值，表示该学生尚未选择主修专业

（2）非空值，这时该值必须是专业关系中某个元组的“专业名”值，表示该学生不可能选一个不存在的专业

参照完整性约束（续）

[例2.2] 中

学生选课（学号，课程号，成绩，选课学期，教学班）

“学号”和“课程号”可能的取值：

（1）学生选课关系中的主属性，不能取空值

（2）只能取相应被参照关系中已经存在的主码值

参照完整性约束（续）

[例2.3] 中

课程（课程号，课程名，学分，先修课）

“先修课”属性值可以取两类值：

（1）空值，表示该门课程不存在先修课

（2）非空值，该值必须是本关系中某个元组的课程号

2.3 关系的完整性

2.3.1 实体完整性

2.3.2 参照完整性

2.3.3 用户定义的完整性

2.3.3 用户定义的完整性

- 针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求
- 关系模型应提供定义和检验这类完整性的机制，以便用统一的系统的方法处理它们，而不需由应用程序承担这一功能

用户定义的完整性（续）

例2.1 “学生”关系中

- 若要求学生不能没有姓名，则可以定义学生“姓名”不能取空值
- “学生选课”关系中“成绩”的取值范围可以定义在0～100之间

第2章 关系数据库

2.4 关系代数

- 关系代数是一种抽象的查询语言，它用对关系的运算来表达查询
- 关系代数
  - 运算对象是关系
  - 运算结果亦为关系
  - 关系代数的运算符：集合运算符和专门的关系运算符
- 传统的集合运算是从关系的“水平”方向即行的角度进行
- 专门的关系运算不仅涉及行而且涉及列

2.4 关系代数

表2.4 关系代数运算符

运算符		含义
集合运算符	∪	并
	-	差
	∩	交
	×	笛卡儿积
专门的关系运算符	σ	选择
	π	投影
	⋈	连接
	÷	除

2.4 关系代数

2.4.1 传统的集合运算

2.4.2 专门的关系运算

（1）并（union）

- ***R*和*S***
  - ■具有相同的目*n*（即两个关系都有*n*个属性）
  - ■相应的属性取自同一个域

- ***R*∪*S***
  - ■仍为*n*目关系，由属于*R*或属于*S*的元组组成

$$R \cup S = \{ t \mid t \in R \vee t \in S \}$$

并（续）

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

RUS

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_2

（2）差（difference）

- ***R*和*S***
  - ■具有相同的目***n***
  - ■相应的属性取自同一个域

- ***R - S***
  - ■仍为***n***目关系，由属于***R***而不属于***S***的所有元组组成

$$R - S = \{ t \mid t \in R \wedge t \notin S \}$$

差（续）

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

R-S

A	B	C
a_1	b_1	c_1

（3） 交（intersection）

- ***R*和*S***
  - ■具有相同的目*n*
  - ■相应的属性取自同一个域
- ***R*∩*S***
  - ■仍为*n*目关系，由既属于*R*又属于*S*的元组组成

$$R \cap S = \{ t \mid t \in R \wedge t \in S \}$$

$$R \cap S = R - (R - S)$$

交（续）

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

R ∩ S

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

（4） 笛卡儿积（Cartesian product）

- 严格地讲应该是广义的笛卡儿积（extended Cartesian product）
- R: n目关系，k_1个元组
- S: m目关系，k_2个元组
- $R \times S$
  - （$n+m$）列元组的集合
    - 元组的前n列是关系R的一个元组
    - 后m列是关系S的一个元组
  - 行：$k_1 \times k_2$个元组
    - $R \times S = \{t_r \, t_s \mid t_r \in R \wedge t_s \in S\}$

笛卡儿积（续）

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

R × S

R.A	R.B	R.C	S.A	S.B	S.C
a_1	b_1	c_1	a_1	b_2	c_2
a_1	b_1	c_1	a_1	b_3	c_2
a_1	b_1	c_1	a_2	b_2	c_1
a_1	b_2	c_2	a_1	b_2	c_2
a_1	b_2	c_2	a_1	b_3	c_2
a_1	b_2	c_2	a_2	b_2	c_1
a_2	b_2	c_1	a_1	b_2	c_2
a_2	b_2	c_1	a_1	b_3	c_2
a_2	b_2	c_1	a_2	b_2	c_1

2.4 关系代数

2.4.1 传统的集合运算

2.4.2 专门的关系运算

2.4.2 专门的关系运算

先引入几个记号

（1） R，t∈R，$t[A_i]$

设关系模式为$R(A_1, A_2, \ldots, A_n)$

它的一个关系设为R

$t \in R$表示t是R的一个元组

$t[A_i]$则表示元组t中相应于属性A_i的一个分量

专门的关系运算（续）

（2） A，t[A]， $\overline{A}$

若$A=\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$，其中$A_{i1}$，$A_{i2}$，…，$A_{ik}$是$A_1$，$A_2$，…，$A_n$中的一部分，则$A$称为属性列或属性组。

$t[A]=(t[A_{i1}], t[A_{i2}], \ldots, t[A_{ik}])$表示元组$t$在属性列$A$上诸分量的集合。

$\overline{A}$则表示$\{A_1, A_2, \ldots, A_n\}$中去掉$\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$后剩余的属性组。

专门的关系运算（续）

（3） $\widehat{t_r t_s}$

R为n目关系，S为m目关系

$t_r \in R$，$t_s \in S$，$\widehat{t_r t_s}$称为元组的连接或元组的串接

$\widehat{t_r t_s}$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

专门的关系运算（续）

（4）象集Z_x

给定一个关系R（X，Z），X和Z为属性组

当$t[X]=x$时，x在R中的象集（images set）定义为：

$$Z_x=\{t[Z]|t \in R，t[X]=x\}$$

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

专门的关系运算（续）

R	
x_1	Z_1
x_1	Z_2
x_1	Z_3
x_2	Z_2
x_2	Z_3
x_3	Z_1
x_3	Z_3

象集举例

- x_1在R中的象集

  Z_{x1} ={Z_1，Z_2，Z_3}，

- x_2在R中的象集

  Z_{x2} ={Z_2，Z_3}，

- x_3在R中的象集

  Z_{x3}={Z_1，Z_3}

专门的关系运算（续）

1. 选择
2. 投影
3. 连接
4. 除

专门的关系运算（续）

学生选课数据库:

学生关系Student、课程关系Course和学生选课关系SC

Student

学号 Sno	姓名 Sname	性别 Ssex	出生日期 Sbirthdate	主修专业 Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	刘晨	女	1999-9-1	计算机科学与技术
20180003	王敏	女	2001-8-1	计算机科学与技术
20180004	张立	男	2000-1-8	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180006	赵明	男	2000-6-12	数据科学与大数据技术
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

专门的关系运算（续）

Course

课程号 Cno	课程名 Cname	学分 Ccredit	先修课 Cpno
81001	程序设计基础与C语言	4	
81002	数据结构	4	81001
81003	数据库系统概论	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

专门的关系运算（续）

SC

学号 Sno	课程号 Cno	成绩 Grade	选课学期 Semester	教学班 Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81002	97	20201	81002-02
20180005	81003	68	20202	81003-01

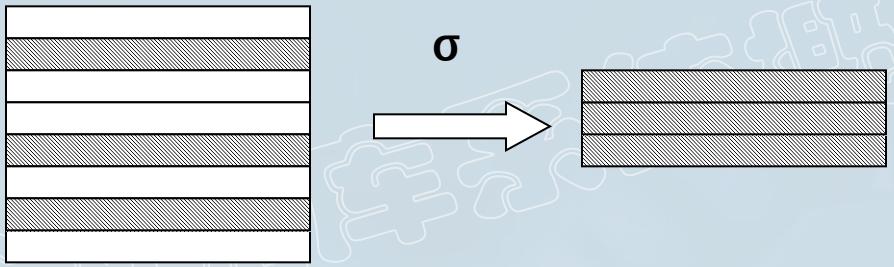
1. 选择（selection）

- 选择又称为限制（restriction）
- 选择运算符的含义
  - 在关系R中选择满足给定条件的诸元组

    $\sigma_F(R) = \{t|t \in R \wedge F(t) = \text{'真'}\}$
  - F：选择条件，是一个逻辑表达式，取值为“真”或“假”
    - 基本形式为：$X_1 \theta Y_1$
    - θ表示比较运算符，它可以是>，≥，<，≤，＝或<>
    - X_1，Y_1是属性名，或为常量，或为简单函数；也可以用它的序号来代替

选择（续）

- 选择运算是从关系***R***中选取使逻辑表达式***F***为真的元组，是从行的角度进行的运算



选择（续）

[例2.4] 查询信息安全专业全体学生。

$\sigma_{Smajor = '信息安全'}(Student)$

结果：

Sno	Sname	Ssex	Sbirthdate	Smajor
20180001	李勇	男	2000-3-8	信息安全

选择（续）

[例2.5]查询2001年之后（包括2001年）出生的学生

$$\sigma_{Sbirthdate >= 2001\text{-}1\text{-}1}(Student)$$

结果：

Sno	Sname	Ssex	Sbirthdate	Smajor
20180003	王敏	女	2001-8-1	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

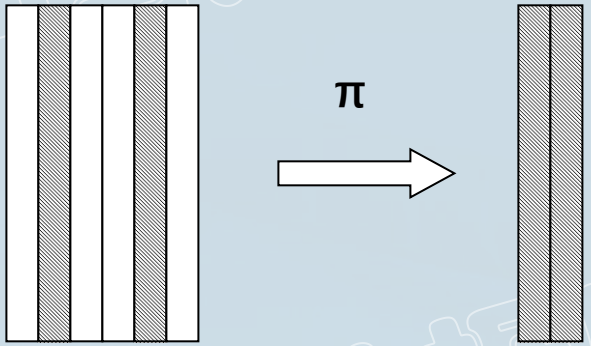
2. 投影（projection）

■从*R*中选择出若干属性列组成新的关系

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

A：*R*中的属性列

■投影操作主要是从列的角度进行运算



■投影之后不仅取消了原关系中的某些列，而且还可能取消某些元组（避免重复行）

投影（续）

❖ [例2.6] 查询学生的姓名和主修专业

即求Student关系上学生姓名和主修专业两个属性上的投影

$\pi_{Sname,Smajor}(Student)$

结果：

学号Sno	专业 Smajor
20180001	信息安全
20180002	计算机科学与技术
20180003	计算机科学与技术
20180004	计算机科学与技术
20180005	信息管理与信息系统
20180006	数据科学与大数据技术
20180007	数据科学与大数据技术

投影（续）

[例2.7] 查询学生关系Student中都主修了哪些专业

即查询关系Student在主修专业属性上的投影

$$\pi_{Smajor}(Student)$$

结果：

专业Smajor
信息安全
计算机科学与技术
信息管理与信息系统
数据科学与大数据技术

3. 连接（join）

- 连接也称为θ连接
- 连接运算的含义

从两个关系的笛卡儿积中选取属性间满足一定条件的元组

$$R \underset{A\theta B}{\bowtie} S = \{ \overset{\frown}{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A]\theta t_s[B] \}$$

- ***A***和***B***：分别为***R***和***S***上度数相等且可比的属性组
- θ：比较运算符

■连接运算从***R***和***S***的广义笛卡儿积***R***×***S***中选取***R***关系在***A***属性组上的值与***S***关系在***B***属性组上的值满足比较关系θ的元组

连接（续）

❖两类常用连接运算

■等值连接（equijoin）

●θ为“＝”的连接运算称为等值连接

●从关系R与S的广义笛卡儿积中选取A、B属性值相等的那些元组，即等值连接为：

$$R \underset{A=B}{\bowtie} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B] \}$$

连接（续）

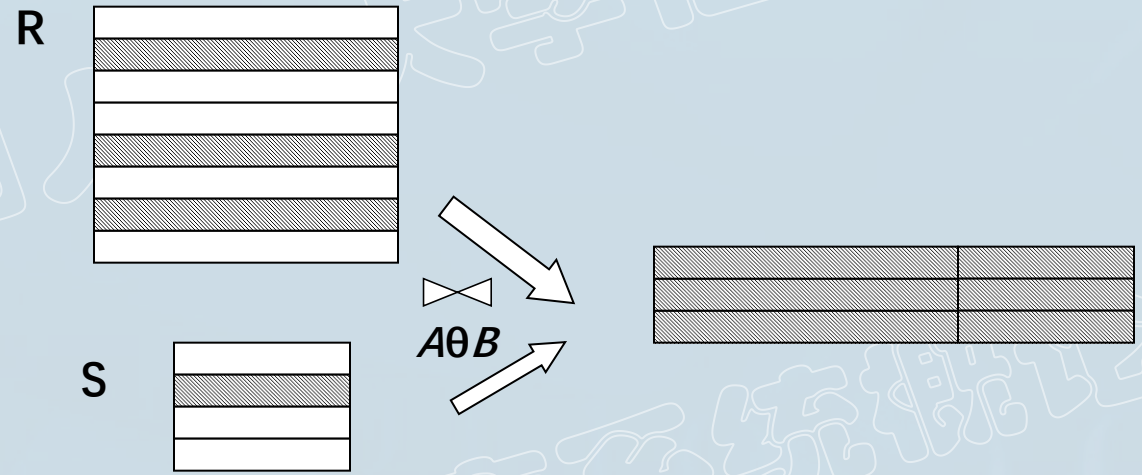
- 自然连接（natural join）
  - 自然连接是一种特殊的等值连接
    - 两个关系中进行比较的分量必须是同名的属性组
    - 在结果中把重复的属性列去掉
  - 自然连接的含义

    R和S具有相同的属性组B，U为R和S的全体属性集合

    $R \bowtie S = \{ \widehat{t_r t_s}\,[U\text{-}B] \mid t_r \in R \wedge t_s \in S \wedge t_r[B] = t_s[B] \}$

连接（续）

❖一般的连接操作是从行的角度进行运算。



自然连接还需要取消重复列，所以是同时从行和列的角度进行运算

连接（续）

❖ [例2.8]关系*R*和关系*S* 如下所示：

R

A	B	C
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

S

B	E
b_1	3
b_2	7
b_3	10
b_3	2
b_2	2

连接（续）

一般连接 $R \underset{C<E}{\bowtie} S$的结果如下：

A	R.B	C	S.B	E
a_1	b_1	5	b_2	7
a_1	b_1	5	b_3	10
a_1	b_2	6	b_2	7
a_1	b_2	6	b_3	10
a_2	b_3	8	b_3	10

连接（续）

等值连接 $R \underset{R.B=S.B}{\bowtie} S$ 的结果如下：

A	R.B	C	S.B	E
a_1	b_1	5	b_1	3
a_1	b_2	6	b_2	7
a_2	b_3	8	b_3	10
a_2	b_3	8	b_3	2

连接（续）

自然连接 $R \bowtie S$的结果如下：

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

连接（续）

❖悬浮元组（dangling tuple）

■两个关系*R*和*S*在做自然连接时，关系*R*中某些元组有可能在*S*中不存在公共属性上值相等的元组，从而造成*R*中这些元组在操作时被舍弃了，这些被舍弃的元组称为悬浮元组。

连接（续）

- 外连接（outer join）
  - 如果把悬浮元组也保存在结果关系中，而在其他属性上填空值(NULL)，就叫做外连接
  - 左外连接(left outer join或left join)
    - 只保留左边关系*R*中的悬浮元组
  - 右外连接(right outer join或right join)
    - 只保留右边关系*S*中的悬浮元组

连接（续）

下图是例2.8中关系*R*和关系*S*的外连接

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL
NULL	b_5	NULL	2

连接（续）

图(b)是例2.8中关系*R*和关系*S*的左外连接,图(c)是右外连接

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL

图(b)

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
NULL	b_5	NULL	2

图(c)

4. 除（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：

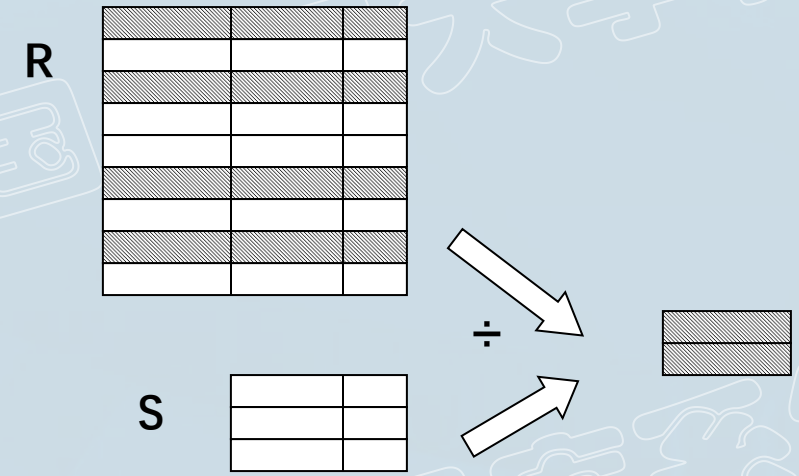
元组在*X*上分量值*x*的象集Y_x包含*S*在*Y*上投影的集合，记作：

$$R \div S=\{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$$

Y_x：*x*在*R*中的象集，$x = t_r[X]$

除运算（续）

- 除操作是同时从行和列角度进行运算



除运算（续）

[例2.9]设关系*R*、*S*，*R* ÷ *S*如下图

R

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

S

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

R÷S

A
a_1

除运算（续）

- 在关系R中，A可以取四个值{a_1，a_2，a_3，a_4}

  a_1的象集为 {(b_1，c_2)，(b_2，c_3)，(b_2，c_1)}

  a_2的象集为 {(b_3，c_7)，(b_2，c_3)}

  a_3的象集为 {(b_4，c_6)}

  a_4的象集为 {(b_6，c_6)}

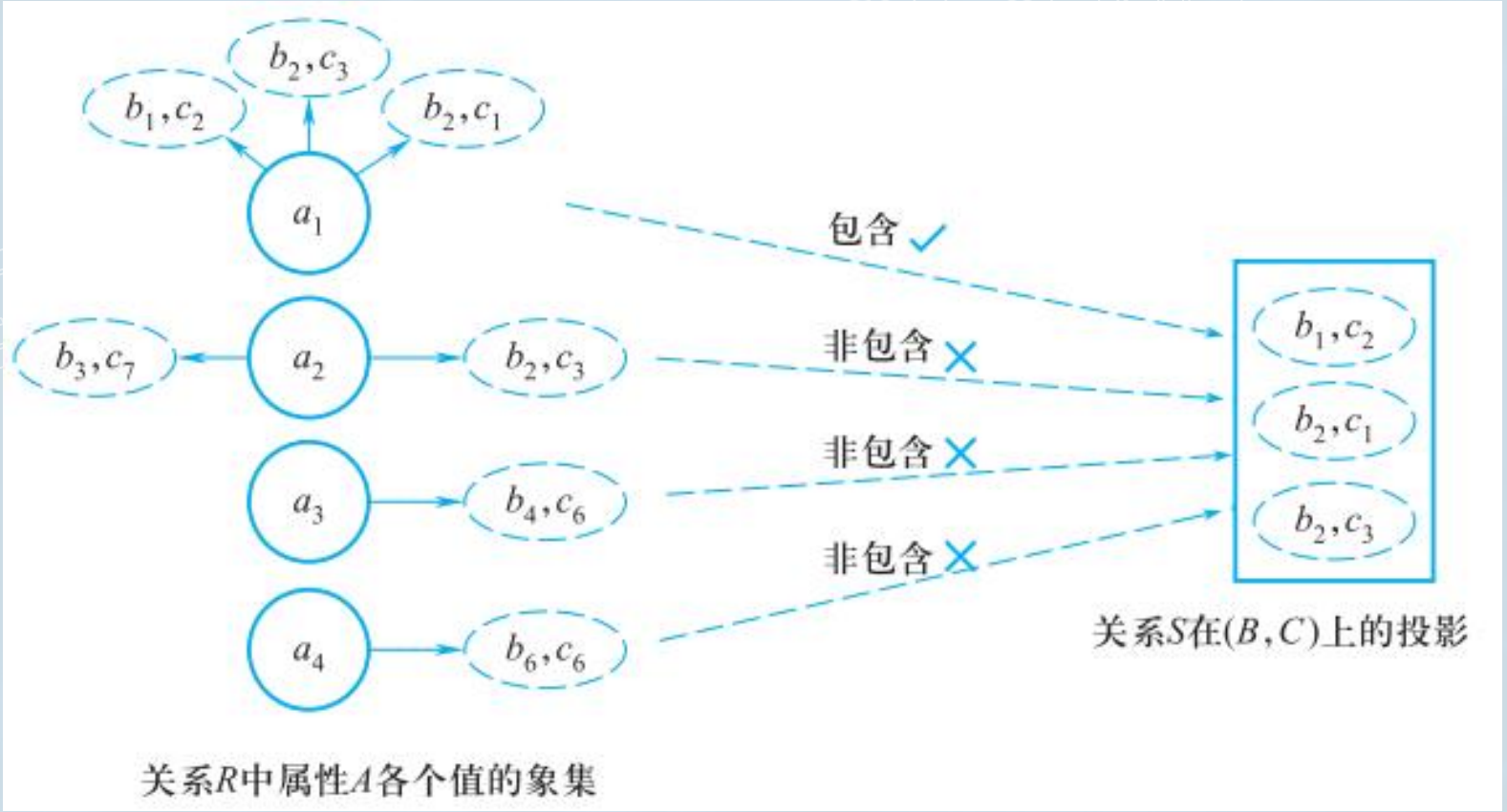
- S在(B，C)上的投影为

  {(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)属性组上的投影

  所以　$R \div S = \{a_1\}$

除运算（续）


b_2,c_3
b_1,c_2
b_2,c_1
a_1
b_3,c_7
a_2
b_2,c_3
a_3
b_4,c_6
a_4
b_6,c_6
包含 ✓
非包含 ✗
非包含 ✗
非包含 ✗
b_1,c_2
b_2,c_1
b_2,c_3
关系S在(B,C)上的投影
关系R中属性A各个值的象集

综合举例

以学生-课程数据库为例

[例2.10]查询至少选修81001号课程和81003号课程的学生号码

首先建立一个临时关系*K*：

Cno
81001
81003

然后求：$\pi_{Sno,Cno}(SC) \div K=\{20180001,20180002\}$

综合举例（续）

[例2.11] 查询选修了81002号课程的学生的学号

$\pi_{Sno}(\sigma_{Cno='81002'}(SC))=\{20180001,20180002,20180003\}$

[例2.12]查询至少选修了一门其直接先修课为81003号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='81003'}(Course) \bowtie SC \bowtie \pi_{Sno,Sname}(Student))$

或

$\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='81003'}(Course) \bowtie SC) \bowtie \pi_{Sno,Sname}(Student))$

[例2.13] 查询选修了全部课程的学生学号和姓名

$\pi_{Sno,Cno}(SC) \div \pi_{Cno}(Course) \bowtie \pi_{Sno,Sname}(Student)$

小结

- 关系代数运算
  - 并、差、笛卡儿积、投影、选择5种基本运算
  - 交、连接、除均可以用这5中基本运算来表达
- 关系代数表达式
  - 关系代数运算经有限次复合后形成的表达式
- 扩展的关系代数
  - 关系的重新命名、查询结果的去重操作、分组操作、 排序操作和聚集函数等

第2章 关系数据库

2.5 关系演算

❖关系演算

以数理逻辑中的谓词演算为基础

❖按谓词变元不同 进行分类

1.元组关系演算

以元组变量作为谓词变元的基本对象

元组关系演算语言ALPHA

2.域关系演算

以元组变量的分量（域变量）作为谓词变元的基本对象

域关系演算语言QBE

2.5 关系演算

2.5.1 *元组关系演算语言ALPHA

2.5.2 *域关系演算语言QBE

2.5.1 元组关系演算语言ALPHA

- 由E.F.Codd提出

  INGRES所用的QUEL语言是参照ALPHA语言研制的

- 语句
  - 6个语句
    - GET，PUT，HOLD，UPDATE，DELETE，DROP
  - 基本格式是
    - 操作语句　工作空间名（表达式）：操作条件

1. 检索操作

（1）简单检索

GET　工作空间名 （表达式1）

[例2.14] 查询所有被选修的课程号码

GET W (SC.Cno)

[例2.15] 查询所有学生的数据。

GET W (Student)

检索操作（续）

（2）限定的检索

格式

GET　工作空间名（表达式1）：操作条件

[例2.16]查询计算机科学与技术专业中2000年之前出生的学生的学号和出生日期

GET W (Student.Sno，Student.Sbirthdate):

Student.Smajor=‘计算机科学与技术’

∧ Student.Sbirhtdate<‘2000-1-1’

检索操作（续）

（3）带排序的检索

格式

GET 工作空间名（表达式1）[：操作条件]

DOWN/UP 表达式2

[例2.17]查询计算机科学与技术专业学生的学号和出生日期，结果按出生日期降序排序

GET W (Student.Sno,Student.Sbirthdate):

Student.Smajor='计算机科学与技术'

DOWN Student.Sbirthdate

检索操作（续）

（4）指定返回元组的条数的检索

GET　工作空间名（定额）（表达式1）

[：操作条件] [DOWN/UP 表达式2]

［例2.18］ 取出一个信息安全专业学生的学号。

GET W (1) (Student.Sno): Student.Smajor='信息安全专业'

在W后括号中的数量就是指定的返回元组的个数

［例2.19］查询选修了81003号课程，成绩前三名的学号及其成绩

GET W (3) (SC.Sno,SC.Grade): SC.Cno='81003' DOWN SC.Grade

检索操作（续）

（5）用元组变量的检索

- 元组变量：在某一关系范围内变化（范围变量range variable）
- 元组变量的用途
  - ① 简化关系名：设一个较短名字的元组变量来代替较长的关系名
  - ② 操作条件中使用量词时必须用元组变量

检索操作（续）

❖ 定义元组变量

■格式：RANGE 关系名 变量名

■一个关系可以设多个元组变量

［例2.20］查询信息安全专业学生的姓名

```
RANGE Student X
GET W (X.Sname): X.Smajor='信息安全'
```

■ALPHA语言用RANGE来说明元组变量

■X是关系Student上的元组变量，用途是简化关系名，即用X代表Student

检索操作（续）

（6） 用存在量词的检索

❖ 操作条件中使用量词时必须用元组变量

[例2.21]查询选修81002号课程的学生姓名

RANGE SC X

GET W (Student.Sname): ∃X(X.Sno=Student.Sno∧X.Cno='81002')

[例2.22] 查询选修了这样课程的学生学号，其直接先修课是81002号课程

RANGE Course CX

GET W (SC.Sno):

∃CX (CX.Cno=SC.Cno∧CX.Cpno='81002')

检索操作（续）

[例2.23]查询至少选修一门其先修课为81002号课程的学生姓名

RANGE Course CX
SC SCX
GET W (Student.Sname):∃SCX (SCX.Sno=Student.Sno∧
∃CX (CX.Cno=SCX.Cno∧CX.Cpno='81002'))

前束范式形式：

RANGE Course CX
SC SCX
GET W (Student.Sname): ∃ SCX∃CX(SCX.Sno=Student.Sno∧
CX.Cno=SCX.Cno∧CX.Cpno='81002')

检索操作（续）

（7）带有多个关系的表达式的检索

[例2.24] 查询成绩为90分以上的学生姓名与课程名称

RANGE SC SCX

GET W (Student.Sname,Course.Cname): ∃ SCX (SCX.Grade≥90

∧SCX.Sno=Student.Sno

∧Course.Cno=SCX.Cno)

检索操作（续）

（8）用全称量词的检索

[例2.25] 查询不选81004号课程的学生姓名

RANGE SC SCX

GET W (Student.Sname): ∀SCX (SCX.Sno≠Student.Sno

∨ SCX.Cno≠'81004')

也可以用存在量词来表示：

RANGE SC SCX

GET W (Student.Sname):¬∃SCX (SCX.Sno=Student.Sno

∧ SCX.Cno='81004')

检索操作（续）

（9）用两种量词的检索

[例2.26] 查询选修了全部课程的学生姓名

RANGE Course CX

SC SCX

GET W (Student.Sname): ∀CX∃SCX (SCX.Sno=Student.Sno

∧ SCX.Cno=CX.Cno)

检索操作（续）

（10）用蕴涵（implication）的检索

[例2.27] 查询最少选修了20180003学生所选课程的学生学号

```
RANGE  Couse  CX
       SC    SCX
       SC    SCY
GET  W  (Student.Sno):
       ∀CX(∃SCX(SCX.Sno='20180003'∧SCX.Cno=CX.Cno)
        ⇨ ∃SCY(SCY.Sno=Student.Sno∧SCY.Cno= CX.Cno))
```

检索操作（续）

（11）聚集函数

常用聚集函数（aggregation function）或内置函数（build-in function）

函数名	功能
COUNT	对元组计数
TOTAL	求总和
MAX	求最大值
MIN	求最小值
AVG	求平均值

关系演算中的聚集函数

检索操作（续）

[例2.28] 查询学生所选的专业的数目

GET W（COUNT(Student.Smajor)）

COUNT函数在计数时会自动排除重复值

[例2.29]查询2020年第二学期选修了81003号课程的平均成绩

GET W (AVG(SC.Grade)) : SC.Cno='81003’

∧SC.Semester='20202'

2. 更新操作

（1） 修改操作

（2） 插入操作

（3） 删除操作

（1）修改操作

修改操作用UPDATE语句实现。其步骤是：

① 用HOLD语句将要修改的元组从数据库中读到工作空间中

HOLD 工作空间名（表达式1）[：操作条件]

HOLD语句是带上并发控制的GET语句

② 用宿主语言修改工作空间中元组的属性值

③ 用UPDATE语句将修改后的元组送回数据库中

UPDATE 工作空间名

修改操作（续）

[例2.30] 把20180004学生从计算机科学与技术专业转到信息管理与信息系统专业

```
HOLD W (Student.Sno, Student Smajor): Student.Sno='20180004'
                    /* 从Student关系中读出20180004学生的数据 */
MOVE '信息管理与信息系统' TO W.Smajor
                    /* 用宿主语言进行修改 */
UPDATE W
                    /*把修改后的元组送回Student关系*/
```

- 如果修改操作涉及两个关系的话，要执行两次HOLD-MOVE-UPDATE操作序列
- 在ALPHA语言中，修改关系主码的操作是不允许的
  - 如果需要修改主码值，先删除该元组，再把具有新主码值的元组插入到关系中

（2）插入操作

插入操作用PUT语句实现。其步骤是：

① 用宿主语言在工作空间中建立新元组

② 用PUT语句把该元组存入指定关系中

PUT 工作空间名 （关系名）

PUT语句只对一个关系操作，即表达式必须为单个关系名

插入操作（续）

[例2.31]学校新开设了一门2学分的课程“计算机组织与结构”，其课程号为81009，直接先行课为81005号课程。插入该课程元组。

```
MOVE '81009' TO W.Cno
MOVE '计算机组织与结构' TO W.Cname
MOVE '2' TO W.Ccredit
MOVE '81005' TO W.Cpno
PUT W (Course)          /*把W中的元组插入指定关系Course中*/
```

（3）删除操作

删除操作用DELETE语句实现

其步骤为：

① 用HOLD语句把要删除的元组从数据库中读到工作空间中

② 用DELETE语句删除该元组

DELETE 工作空间名

删除操作（续）

[例2.32]20180007学生因故退学，删除该学生元组

HOLD W (Student): Student.Sno='20180007’

DELETE W

[例2.33]将学号20180001改为20180010

HOLD W (Student): Student.Sno='20180001’

DELETE W

MOVE '20180010' TO W.Sno

MOVE '李勇' TO W.Sname

MOVE '男' TO W.Ssex

MOVE '2000-3-8' TO W.Sbirthdate

MOVE '信息安全' TO W.Smajor

PUT W (Student)

删除操作（续）

[例2.34]删除全部学生

HOLD W (Student)

DELETE W

由于SC关系与Student关系之间具有参照关系，为保证参照完整性，删除Student中元组时相应地要删除SC中的元组：

HOLD W (SC)

DELETE W

小结：元组关系演算语言ALPHA

- ❖检索操作 GET

  GET 工作空间名 [（定额）]（表达式1）

  [：操作条件] [DOWN/UP 表达式2]

- ❖插入操作
  - ■建立新元组--PUT
- ❖修改操作
  - ■HOLD--修改--UPDATE
- ❖删除操作
  - ■HOLD—DELETE

二维码 2.2

元组演算表达式

2.5 关系演算

2.5.1 *元组关系演算语言ALPHA

2.5.2 *域关系演算语言QBE

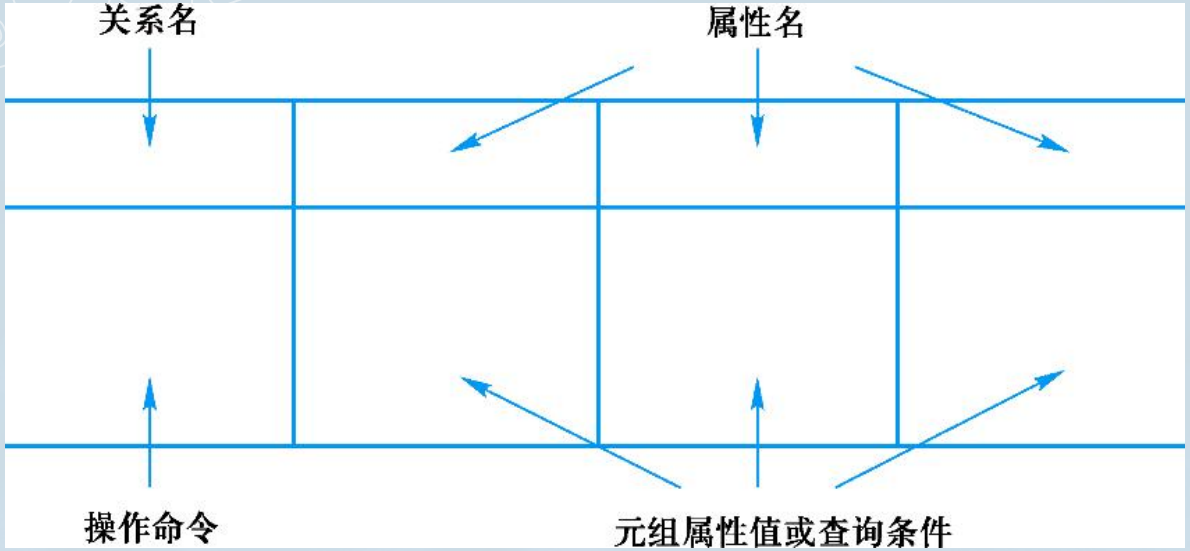
2.5.2 域关系演算语言QBE

- 域关系演算语言
  - 由M.M.Zloof提出
  - 以元组变量的分量（即域变量）作为谓词变元的基本对象
- QBE：Query By Example
  - 基于屏幕表格的查询语言
  - 查询要求：以填写表格的方式构造查询
  - 查询结果：以表格形式显示
  - 用示例元素(域变量)来表示查询结果可能的情况

QBE操作框架

QBE中用示例元素来表示查询结果可能的情况，示例元素实质上就是域变量。

QBE操作框架如图所示



1. 检索操作

（1）简单查询

[例2.35]求信息安全专业全体学生的姓名

操作步骤为：

①用户提出要求；

②屏幕显示空白表格；

简单查询（续）

③用户在最左边一栏输入要查询的关系名**Student**

Student					

④系统显示该关系的属性名

Student	Sno	Sname	Ssex	Sbirthdate	Smajor

简单查询（续）

⑤用户在上面构造查询要求

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
		P.T			信息安全

- T是示例元素，即域变量，一定要加下划线。信息安全是查询条件，不用加下划线
- P.是操作符，表示打印（Print），实际上是显示。
- 查询条件：可使用比较运算符>，≥，<，≤，＝和≠，其中＝可以省略
- 示例元素是这个域中可能的一个值

简单查询（续）

对于例2.35，可如下构造查询要求：

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
		P.刘晨			信息安全

⑥屏幕显示查询结果：

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
		李勇			信息安全

简单查询（续）

[例2.36] 查询全体学生的全部数据

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	P.20180002	P.刘晨	P.女	P.1999-9-1	P.计算机科学与技术

显示全部数据也可以简单地把P.操作符作用在关系名上

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
P.					

（2）条件查询

[例2.37] 求出生日期晚于1999-9-1的学生的学号

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	P.20180002			>1999-9-1	

条件查询（与条件）

[例2.38] 求计算机科学与技术专业出生日期晚于1999-9-1的学生的学号。

- 方法①：把两个条件写在同一行上

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	P.20180002			>1999-9-1	计算机科学与技术

条件查询（与条件）

方法②：把两个条件写在不同行上，但使用相同的示例元素值

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	P. 20180002				计算机科学与技术
	P. 20180002			>1999-9-1	

条件查询（或条件）

[例2.39]查询计算机科学系或者出生日期晚于1999-9-1的学生的学号

两个条件写在不同行上，并且使用不同的示例元素值，即表示条件的“或”

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	P. 20180002				计算机科学与技术
	P. 20180003			>1999-9-1	

条件查询（与条件）

[例2.40] 查询既选修了81001号课程又选修了81002号课程的学生的学号

SC	Sno	Cno	Grade	Semester	Teachingclass
	P.20180002	81001			
	P.20180002	81002			

条件查询（多个关系）

[例2.41] 查询选修了81001号课程的学生姓名

本查询涉及两个关系：SC和Student，通过相同的连接属性值把多个关系连接起来。这里示例元素Sno是连接属性，其值在两个表中要相同

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	20180002	P.李勇			

SC	Sno	Cno	Grade	Semester	Teachingclass
	20180002	81001			

条件查询（非条件）

[例2.42] 查询未选修81001号课程的学生姓名

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	20180002	P.李勇			
	20180002	P.李勇			

SC	Sno	Cno	Grade	Semester	Teachingclass
¬	20180002	81001			
¬	20180002	_			

条件查询（续）

[例2.43] 查询有两个人以上选修的课程号

Sc	Sno	Cno	Grade	Semester	Teachingclass
	20180002	P.81001			
	¬ 20180002	81001			

思路：查询这样的课程81001，它不仅被20180002选修
而且也被另一个学生（¬20180002）选修了

（3）聚集函数

常用聚集函数：

函数名	功能
CNT	对元组计数
SUM	求总和
AVG	求平均值
MAX	求最大值
MIN	求最小值

QBE中的聚集函数

聚集函数（续）

[例2.44]查询信息安全专业学生的总人数

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	P.CNT			.	信息安全

（4）对查询结果排序

- 升序排序：
  - 对查询结果按某个属性值的升序排序，只需在相应列中填入“AO.”
- 降序排序：
  - 按降序排序则填“DO.”
- 多列排序：
  - 如果按多列排序，用“AO(i).”或“DO(i).”表示，其中i为排序的优先级，i值越小，优先级越高

对查询结果排序（续）

[例2.45]查询全体男生的姓名，要求查询结果按专业名升序排序，对同一专业的学生按出生日期降序排序

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
		P.李勇	男	DO(2).	AO(1).

2. 更新操作

（1）修改操作

[例2.46]学号为20180001学生的出生日期改为'2001-3-8'

方法①：将操作符“U.”放在值上

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
	20180001			U.2001-3-8	

修改操作（续）

方法②： 将操作符“U.”放在关系上

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
U.	20180001			2001-3-8	

码2018001标明要修改的元组。

“U.”标明所在的行是修改后的新值。

由于主码不能修改，系统不会混淆要修改的属性值。

修改操作（续）

[例2.47]把学号为20180004的学生在2019年第2学期选修81001号课程的成绩增加6分

SC	Sno	Cno	Grade	Semester	Teachingclass
	20180004	81001	56	20192	
U.	20180004		56+6		

操作涉及表达式，必须将操作符“U.”放在关系上

修改操作（续）

[例2.48]将2019年第2学期选修81001号课程的学生成绩都增加5分

SC	Sno	Cno	Grade	Semester	Teachingclass
	20180001	81001	70	20192	
U.	20180001		70+5	20192	

(2) 插入操作

[例2.49]把计算机科学与技术专业的学生高大卫，男，学号20190006，出生日期'2001-6-28'存入数据库中

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
I.	20190006	高大卫	男	2001-6-28	计算机科学与技术

（3） 删除操作

例[2.50] 删除学号为20180006的学生

Student	Sno	Sname	Ssex	Sbirthdate	Smajor
D.	20180006				

SC关系与Student关系之间具有参照关系，删除20180006学生后，还应删除20180006学生选修的全部课程

SC	Sno	Cno	Grade	Semester	Teachingclass
D.	20180006				

第2章 关系数据库

本章小结

- 关系数据库系统是目前使用最广泛的数据库系统
- 关系模型与层次、网状模型最重要的区别：
  - 关系模型只有“表”这一种数据结构
  - 层次、网状模型还有其他数据结构，以及对这些数据结构的操作

本章小结（续）

- ❖关系数据结构
  - ■ 关系
    - ● 域
    - ● 笛卡儿积
    - ● 关系
      - ➢ 关系，属性，元组
      - ➢ 候选码，主码，主属性
      - ➢ 基本关系的性质
  - ■ 关系模式
  - ■ 关系数据库
  - ■关系模型的存储结构

本章小结（续）

- 关系操作
  - 查询
    - 选择、投影、连接、除、并、交、差、笛卡儿积等
  - 数据更新
    - 插入、删除、修改

本章小结（续）

- ❖关系的完整性
  - ■实体完整性
  - ■参照完整性
    - ●外码
  - ■用户定义的完整性

本章小结（续）

- 关系数据语言
  - 关系代数语言
  - 关系演算语言
    - 元组关系演算语言 ALPHA
    - 域关系演算语言 QBE