数据库系统概论
An Introduction to Database System

第二章 关系数据库

关系数据库简介

- 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”，《Communication of the ACM》,1970
  - 之后，提出了关系代数和关系演算的概念
  - 1972年提出了关系的第一、第二、第三范式
  - 1974年提出了关系的**BC**范式

第二章 关系数据库

2.1 关系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

2.1.1 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. 笛卡尔积（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
  - 长度指定长度的字符串集合
  - {‘男’，‘女’}
  - ……

2. 笛卡尔积（Cartesian Product）

- 笛卡尔积

  给定一组域D_1，D_2，…，D_n，这些域中可以有相同的。

  D_1，D_2，…，D_n的笛卡尔积为：

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

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

  - 所有域的所有取值的一个组合
  - 不能重复

笛卡尔积（续)

- 元组（Tuple）
  - 笛卡尔积中每一个元素（d_1，d_2，…，d_n）叫作一个n元组（n-tuple）或简称元组(Tuple)
  - (张清玫，计算机专业，李勇)、(张清玫，计算机专业，刘晨)等都是元组
- 分量（Component）
  - 笛卡尔积元素（d_1，d_2，…，d_n）中的每一个值d_i叫作一个分量
  - 张清玫、计算机专业、李勇、刘晨等都是分量
- 基数（Cardinal number）
  - 若D_i（i=1，2，…，n）为有限集，其基数为m_i（i=1，2，…，n），则$D_1 \times D_2 \times \cdots \times D_n$的基数$M$为：

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

- 笛卡尔积的表示方法
  - 笛卡尔积可表示为一个二维表
  - 表中的每行对应一个元组，表中的每列对应一个域

表 2.1　D_1，D_2，D_3 的笛卡尔积

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
张清玫	计算机专业	王敏
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
张清玫	信息专业	王敏
刘逸	计算机专业	李勇
刘逸	计算机专业	刘晨
刘逸	计算机专业	王敏
刘逸	信息专业	李勇
刘逸	信息专业	刘晨
刘逸	信息专业	王敏

3. 关系（Relation）

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）

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

表 2.2　SAP 关系

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
刘逸	信息专业	王敏

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

关系（续）

6) 码

候选码（Candidate key）

若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码

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组是这个关系模式的候选码，称为全码（All-key）

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）

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）

关系（续）

- *D*1，*D*2，…，*D*n的笛卡尔积的某个子集才有实际含义

例：表2.1 的笛卡尔积没有实际意义

取出有实际意义的元组来构造关系

关系：SAP(SUPERVISOR，SPECIALITY，POSTGRADUATE)

假设：导师与专业：1:1，　　导师与研究生：1:n

主码：POSTGRADUATE（假设研究生不会重名）

SAP关系可以包含三个元组

{ (张清玫，计算机专业，李勇)，

(张清玫，计算机专业，刘晨)，

(刘逸，信息专业，王敏) }

关系（续）

7) 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

查询表

查询结果对应的表

视图表

由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据

关系（续）

8)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 任意两个元组的候选码不能相同

⑤ 行的顺序无所谓，行的次序可以任意交换

⑥ 分量必须取原子值

这是规范条件中最基本的一条

基本关系的性质(续)

表2.3 非规范化关系

SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	信息专业	李勇	刘晨
刘逸	信息专业	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

2.1.2 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（Relation Schema）是型
- 关系是值
- 关系模式是对关系的描述
  - 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
  - 元组语义以及完整性约束条件
  - 属性间的数据依赖关系集合

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　　属性组*U*中属性所来自的域

DOM　属性向域的映象集合

F　　属性间的数据依赖关系集合

例：导师和研究生出自同一个域——人，

取不同的属性名，并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-PERSON）= DOM（POSTGRADUATE-PERSON）=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

注：域名及属性向域的映象常常直接说明为属性的类型、长度

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

2.1.3 关系数据库

- 关系数据库
  - 在一个给定的应用领域中，所有关系的集合构成一个关系数据库
- 关系数据库的型与值
  - 关系数据库的型：关系数据库模式

    对关系数据库的描述。
  - 关系数据库模式包括
    - 若干域的定义
    - 在这些域上定义的若干关系模式
  - 关系数据库的值：关系模式在某一时刻对应的关系的集合，简称为关系数据库

第二章 关系数据库

2.1 关系模型概述

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结

2.2.1基本关系操作

- 常用的关系操作
  - 查询：选择、投影、连接、除、并、交、差
  - 数据更新：插入、删除、修改
  - 查询的表达能力是其中最主要的部分
  - 选择、投影、并、差、笛卡尔基是5种基本操作
- 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，一次一集合的方式

2.2.2 关系数据库语言的分类

- 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
- 具有关系代数和关系演算双重特点的语言
  - 代表：SQL（Structured Query Language）

第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结

2.3 关系的完整性

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.1 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关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.2 实体完整性

规则2.1 实体完整性规则（Entity Integrity）

若属性*A*是基本关系*R*的主属性，则属性*A*不能取空值

例：

SAP(SUPERVISOR，SPECIALITY，POSTGRADUATE)

POSTGRADUATE：

主码（假设研究生不会重名）

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) 主码中的属性即主属性不能取空值。

主属性取空值，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关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.3 参照完整性

1. 关系间的引用
2. 外码
3. 参照完整性规则

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，因此可能存在着关系与关系间的引用。

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）

- 学生关系引用了专业关系的主码“专业号”。
- 学生关系中的“专业号”值必须是确实存在的专业的专业号 ，即专业关系中有该专业的记录。

关系间的引用(续)

例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，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

例3 学生实体及其内部的一对多联系

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	

- “学号”是主码，“班长”是外码，它引用了本关系的“学号”
- “班长” 必须是确实存在的学生的学号

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。如果F与基本关系*S*的主码K_s相对应，则称F是基本关系*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1：学生关系的“专业号”与专业关系的主码“专业号”相对应
  - “专业号”属性是学生关系的外码
  - 专业关系是被学生关系 $\xrightarrow{\text{专业号}}$ 专业关系 参照关系

(a)

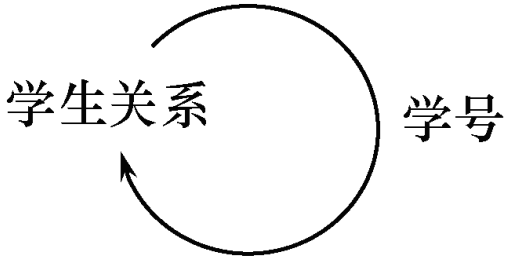
- 例2：选修关系的“学号” 与学生关系的主码“学号”相对应
  选修关系的“课程号”与课程关系的主码“课程号”相对应
  - “学号”和“课程号”是选修关系的外码
  - 学生关系和课程关系均为被
  - 选修关系为参照关系

学生关系 $\xleftarrow{\text{学号}}$ 选修关系 $\xrightarrow{\text{课程号}}$ 课程关系

(b)

外码(续)

- 例3：“班长”与本身的主码“学号”相对应
  - “班长”是外码
  - 学生关系既是参照关系也是被参照关系



(c)

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K_s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，以便于识别

3. 参照完整性规则

规则2.2 参照完整性规则

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K_s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

参照完整性规则(续)

- 例1：学生关系中每个元组的“专业号”属性只取两类值：
  - 空值，表示尚未给该学生分配专业
  - 非空值，这时该值必须是专业关系中某个元组的“专业号”值，表示该学生不可能分配一个不存在的专业
- 例2：选修（学号，课程号，成绩）

  “学号”和“课程号”可能的取值 ：
  - 选修关系中的主属性，不能取空值
  - 只能取相应被参照关系中已经存在的主码值

参照完整性规则(续)

例3：学生（学号，姓名，性别，专业号，年龄，班长）

“班长”属性值可以取两类值：

（1）空值，表示该学生所在班级尚未选出班长

（2）非空值，该值必须是本关系中某个元组的学号值

关系的完整性(续)

2.3.1关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.4 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，而不要由应用程序承担这一功能

用户定义的完整性(续)

例：　课程(课程号，课程名，学分)

- “课程号”属性必须取唯一值
- 非主属性“课程名”也不能取空值
- “学分”属性只能取值{1，2，3，4}

第二章 关系数据库

2.4 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算

概 述

表2.4 关系代数运算符

运算符		含义	运算符		含义
集合运算符	∪ - ∩ ×	并 差 交 笛卡尔积	比较运算符	> ≥ < ≤ = <>	大于 大于等于 小于 小于等于 等于 不等于
专门的关系运算符	σ π ⋈ ÷	选择 投影 连接 除	逻辑运算符	¬ ∧ ∨	非 与 或

2.4 关系代数

- 概述
- 传统的集合运算
- 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
- $R \cup S$
  - 仍为n目关系，由属于R或属于S的元组组成

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

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

$R \cup S$

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

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

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
- $R \cap S$
  - 仍为n目关系，由既属于R又属于S的元组组成

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

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

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

$R \cap S$

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

4. 笛卡尔积（Cartesian Product）

- 严格地讲应该是广义的笛卡尔积（Extended Cartesian Product）
- R: n目关系，k_1个元组
- S: m目关系，k_2个元组
- $R \times S$
  - 列：（$n+m$）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
    - $R \times S = \{t_r \widehat{\ } t_s \mid t_r \in R \wedge t_s \in S\}$

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

$R \times S$

$R.A$	$R.B$	$R.C$	$S.A$	$S.B$	$S.C$
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

- 概述
- 传统的集合运算
- 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

（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。

$\widehat{t_r t_s}$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。

（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）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} ={Z1，Z2，Z3}，

- x_2在R中的象集

  Z_{x2} ={Z2，Z3}，

- x_3在R中的象集

  Z_{x3}={Z1，Z3}

专门的关系运算(续)

- 选择
- 投影
- 连接
- 除

专门的关系运算(续)

4) 学生-课程数据库:

学生关系Student、课程关系Course和选修关系SC

Student (a)

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
200215121	李勇	男	20	CS
200215122	刘晨	女	19	IS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

Course (b)

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

SC (c)

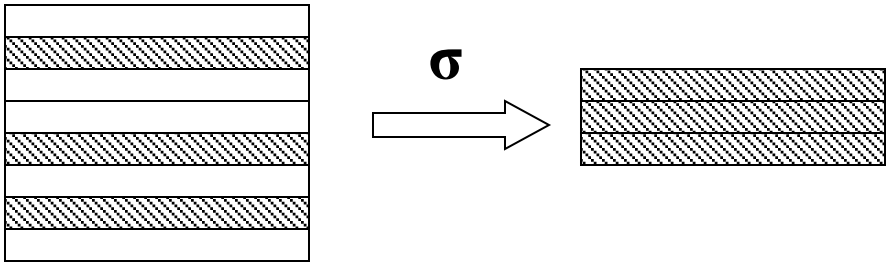
学号 Sno	课程号 Cno	成绩 Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

1. 选择（Selection）

- 1) 选择又称为限制（Restriction）
- 2) 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，基本形式为：

    $X_1 \theta Y_1$
- 3) 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1] 查询信息系（IS系）全体学生

$\sigma_{Sdept = 'IS'}(Student)$

或 $\sigma_{5 ='IS'}(Student)$

结果：

Sno	Sname	Ssex	Sage	Sdept
200215122	刘晨	女	19	IS
200215125	张立	男	19	IS

[例2] 查询年龄小于20岁的学生

$\sigma_{Sage < 20}(Student)$

或 $\sigma_{4 < 20}(Student)$

结果：

Sno	Sname	Ssex	Sage	Sdept
200215122	刘晨	女	19	IS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

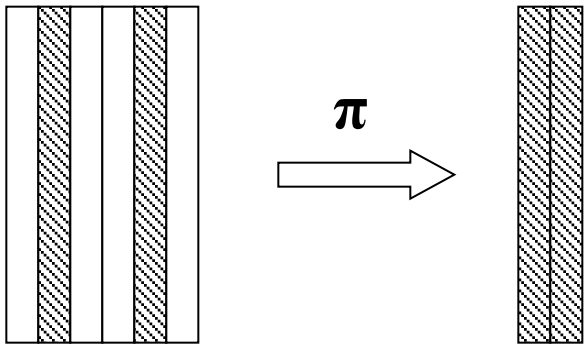
2. 投影（Projection）

- 1）投影运算符的含义
  - 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

- 2）投影操作主要是从列的角度进行运算



  - 但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3] 查询学生的姓名和所在系

  即求Student关系上学生姓名和所在系两个属性上的投影

  $\pi_{Sname，Sdept}(Student)$

  或 $\pi_{2，5}(Student)$

结果：

Sname	Sdept
李勇	CS
刘晨	IS
王敏	MA
张立	IS

- [例4] 查询学生关系Student中都有哪些系

  $\pi_{Sdept}(Student)$

结果：

Sdept
CS
IS
MA

3. 连接（Join）

- 1）连接也称为θ连接
- 2）连接运算的含义

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

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

- 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的广义笛卡尔积$R \times S$中选取（R关系）在A属性组上的值与（S关系）在B属性组上值满足比较关系θ的元组

连接(续)

- 3）两类常用连接运算
  - 等值连接（equijoin）
    - 什么是等值连接

      θ为“＝”的连接运算称为等值连接
    - 等值连接的含义

      从关系R与S的广义笛卡尔积中选取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)

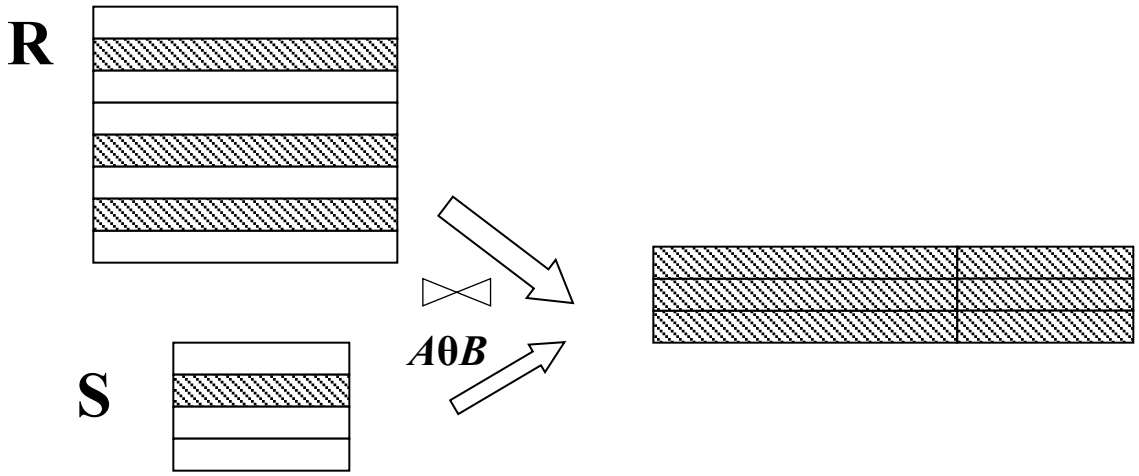
- 自然连接（Natural join）
  - 自然连接是一种特殊的等值连接
    - 两个关系中进行比较的分量必须是相同的属性组
    - 在结果中把重复的属性列去掉
  - 自然连接的含义

    R和S具有相同的属性组B

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

连接(续)

- 4）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5]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_5	2

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

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

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

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)

- 外连接
  - 如果把舍弃的元组也保存在结果关系中，而在其他属性上填空值(Null)，这种连接就叫做外连接（OUTER JOIN）。
- 左外连接
  - 如果只把左边关系*R*中要舍弃的元组保留就叫做左外连接(LEFT OUTER JOIN或LEFT JOIN)
- 右外连接
  - 如果只把右边关系*S*中要舍弃的元组保留就叫做右外连接(RIGHT OUTER JOIN或RIGHT JOIN)。

连接(续)

下图是例5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

(a) 外连接

图(b)是例5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

(b) 左外连接

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
NULL	b_5	NULL	2

(c) 右外连接

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：

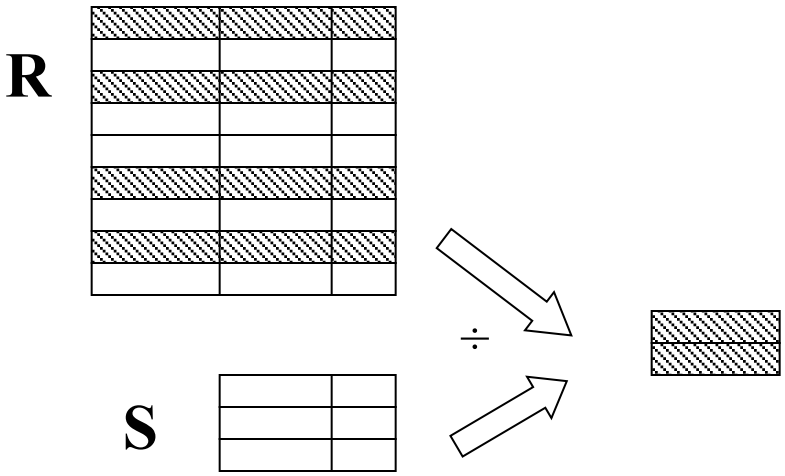
元组在X上分量值x的象集Y_x包含S在Y上投影的集合，记作：

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

Y_x：x在R中的象集，$x = t_r[X]$

除(续)

- 除操作是同时从行和列角度进行运算



除(续)

[例6]设关系R、S分别为下图的(a)和(b)，$R \div S$的结果为图(c)

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

(a)

S

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

(b)

$R \div S$

A
a_1

(c)

分析

- 在关系R中，A可以取四个值{a1，a2，a3，a4}

  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

  $\{(b1, c2), (b2, c1), (b2, c3)\}$

- 只有a_1的象集包含了S在(B, C)属性组上的投影

所以　　$R \div S = \{a_1\}$

5. 综合举例

以学生-课程数据库为例 (P56)

[例7] 查询至少选修1号课程和3号课程的学生号码

首先建立一个临时关系*K*：

Cno
1
3

然后求：$\pi_{Sno,Cno}(SC) \div K$

Sno	Cno
200215121	1
200215121	2
200215121	3
200215122	2
200215122	3

200215121象集 {1，2，3}

200215122象集 {2，3}

K={1，3}

于是：$\pi_{Sno,Cno}(SC) \div K$= {200215121}

综合举例(续)

[例8] 查询选修了2号课程的学生的学号。

π_{Sno}（$\sigma_{Cno='2'}$（SC））＝｛200215121，200215122｝

[例9] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='5'}(Course \bowtie SC \bowtie Student))$

或 $\pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno，Sname}(Student))$

或 $\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(Course) \bowtie SC) \bowtie \pi_{Sno，Sname}(Student))$

[例10] 查询选修了全部课程的学生号码和姓名。

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

小结

- 关系代数运算
  - 关系代数运算
    并、差、交、笛卡尔积、投影、选择、连接、除
  - 基本运算
    并、差、笛卡尔积、投影、选择
  - 交、连接、除
    可以用5种基本运算来表达
    引进它们并不增加语言的能力，但可以简化表达
- 关系代数表达式
  - 关系代数运算经有限次复合后形成的式子
- 典型关系代数语言
  - ISBL（Information System Base Language）
    - 由IBM United Kingdom研究中心研制
    - 用于PRTV（Peterlee Relational Test Vehicle）实验系统

第二章 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：

  以域变量作为谓词变元的基本对象

  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

  检索语句

  - GET

  更新语句

  - PUT，HOLD，UPDATE，DELETE，DROP

一、检索操作

- 语句格式：

GET　工作空间名 [（定额）]（表达式1）

[：操作条件] [DOWN/UP 表达式2]

定额：规定检索的元组个数

- 格式：　数字

表达式1：指定语句的操作对象

- 格式：

关系名| 关系名. 属性名| 元组变量. 属性名| 集函数 [，…]

操作条件：将操作结果限定在满足条件的元组中

- 格式：　逻辑表达式

表达式2：指定排序方式

- 格式：　关系名. 属性名| 元组变量. 属性名[，…]

一、检索操作

（1）简单检索

GET　工作空间名　（表达式1）

[例1] 查询所有被选修的课程号码。

GET W (SC.Cno)

[例2] 查询所有学生的数据。

GET W (Student)

（2）限定的检索

格式

GET　工作空间名（表达式1）：操作条件

[例3]查询信息系(IS)中年龄小于20岁的学生的学号和年龄

GET W (Student.Sno，Student.Sage):

Student.Sdept='IS'∧ Student.Sage<20

（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4]查询计算机科学系(CS)学生的学号、年龄，结果按年龄降序排序

GET W (Student.Sno，Student.Sage):

Student.Sdept='CS ‘ DOWN Student.Sage

（4）带定额的检索

格式

GET 工作空间名（定额）（表达式1）
[：操作条件] [DOWN/UP 表达式2]

[例5] 取出一个信息系学生的学号。

GET W (1) (Student.Sno):
Student.Sdept='IS'

[例6] 查询信息系年龄最大的三个学生的学号及其年龄，结果按年龄降序排序。

GET W (3) (Student.Sno，Student.Sage):
Student.Sdept='IS' DOWN Student.Sage

（5）用元组变量的检索

- 元组变量的含义
  - 表示可以在某一关系范围内变化（也称为范围变量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。
- 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

(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8] 查询选修2号课程的学生名字。

RANGE SC X

GET W (Student.Sname):

∃X(X.Sno=Student.Sno∧X.Cno='2')

[例9] 查询选修了这样课程的学生学号，其直接先行课是6号课程。

RANGE Course CX

GET W (SC.Sno):

∃CX (CX.Cno=SC.Cno∧CX.Pcno='6')

用存在量词的检索(续)

[例10]查询至少选修一门其先行课为6号课程的学生名字

RANGE Course CX

SC SCX

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

∃CX (CX.Cno=SCX.Cno∧CX.Pcno='6'))

前束范式形式：

GET W (Student.Sname):

∃SCX∃CX (SCX.Sno=Student.Sno∧

CX.Cno=SCX.Cno∧CX.Pcno='6')

（7）带有多个关系的表达式的检索

[例11] 查询成绩为90分以上的学生名字与课程名字。

RANGE SC SCX

GET W(Student.Sname，Course.Cname):

∃SCX (SCX.Grade≥90 ∧

SCX.Sno=Student.Sno∧

Course.Cno=SCX.Cno)

（8）用全称量词的检索

[例12] 查询不选1号课程的学生名字

RANGE SC SCX

GET W (Student.Sname):

∀ SCX (SCX.Sno≠Student.Sno∨SCX.Cno≠'1')

用存在量词表示：

RANGE SC SCX

GET W (Student.Sname):

¬∃SCX (SCX.Sno=Student.Sno∧SCX.Cno='1')

（9）用两种量词的检索

[例13] 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)

（10）用蕴函（Implication）的检索

[例14]　查询最少选修了200215122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= ‘200215122'∧SCX.Cno=CX.Cno)

⇨ ∃SCY(SCY.Sno=Student.Sno∧

SCY.Cno= CX.Cno))

（11）聚集函数

常用聚集函数（Aggregation function）或内部函数（Build-in function）

关系演算中的聚集函数

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

[例15] 查询学生所在系的数目。

GET W (COUNT(Student.Sdept))

COUNT函数在计数时会自动排除重复值。

[例16] 查询信息系学生的平均年龄

GET W (AVG(Student.Sage): Student.Sdept='IS')

二、更新操作

(1) 修改操作

(2) 插入操作

(3) 删除操作

（1）修改操作步骤

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

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17] 把200215121学生从计算机科学系转到信息系。

HOLD W (Student.Sno， Student.Sdetp):

Student.Sno=‘200215121'

（从Student关系中读出95007学生的数据）

MOVE 'IS' TO W.Sdept

（用宿主语言进行修改）

UPDATE W

（把修改后的元组送回Student关系）

（2）插入操作

步骤

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，关系演算中的聚集函数

插入操作(续)

[例18]　　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8，直接先行课为6号课程。插入该课程元组

MOVE '8' TO W.Cno

MOVE '计算机组织与结构' TO W.Cname

MOVE '6' TO W.Cpno

MOVE '2' TO W.Ccredit

PUT W (Course)

（3）删除操作

步骤

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

[例19] 200215125学生因故退学，删除该学生元组

HOLD W (Student): Student.Sno= ‘200215125'

DELETE W

删除操作(续)

[例20] 将学号200215121改为200215126

```
HOLD  W  (Student): Student.Sno= ‘200215121'
DELETE  W
MOVE   ‘200215126'  TO  W.Sno
MOVE  '李勇'    TO  W.Sname
MOVE   '男'  TO  W.Ssex
MOVE   '20 ‘  TO  W.Sage
MOVE  'CS'  TO  W.Sdept
PUT  W  (Student)
```

[例21] 删除全部学生

```
HOLD  W  (Student)
DELETE  W
```

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

2.5 关 系 演 算

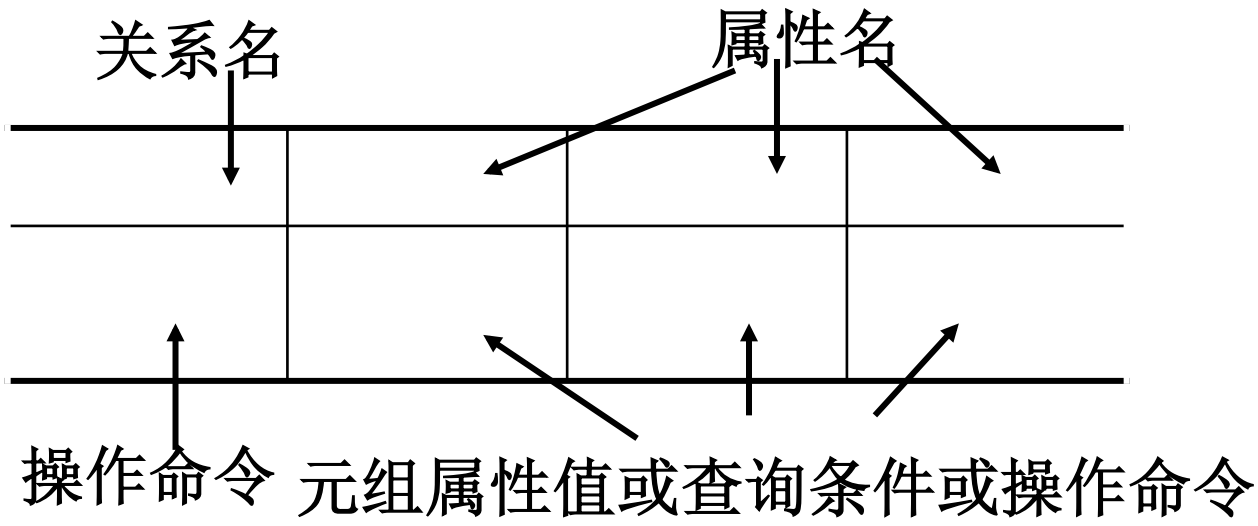
- 2.5.1 元组关系演算语言ALPHA
- 2.5.2 域关系演算语言QBE

2.5.2 域关系演算语言QBE

- 一种典型的域关系演算语言
  - 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
  - 用示例元素(域变量)来表示查询结果可能的情况
  - 查询结果：以表格形式显示

QBE操作框架



一、检索操作

1.简单查询

[例1]求信息系全体学生的姓名

操作步骤为：

（1）用户提出要求；

（2）屏幕显示空白表格；

（3）用户在最左边一栏输入要查询的关系名Student；

Student					

简单查询（续）

（4）系统显示该关系的属性名

Student	Sno	Sname	Ssex	Sage	Sdept

（5）用户在上面构造查询要求

Student	Sno	Sname	Ssex	Sage	Sdept
		P.李勇			IS

- 李勇是示例元素，即域变量

（6）屏幕显示查询结果

Student	Sno	Sname	Ssex	Sage	Sdept
		李勇 张立			IS

构造查询的几个要素

- 示例元素　　即域变量　　一定要加下划线

  示例元素是这个域中可能的一个值，它不必是查询结果中的元素

- 打印操作符P.　　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

简单查询（续）

[例2] 查询全体学生的全部数据

Student	Sno	Sname	Ssex	Sage	Sdept
	P.200215121	P.李勇	P.男	P.20	P.CS

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	Sage	Sdept
P.					

[例3] 求年龄大于19岁的学生的学号

Student	Sno	Sname	Ssex	Sage	Sdept
	P.200215121			>19	

条件查询（与条件）

[例4] 求计算机科学系年龄大于19岁的学生的学号。

方法(1)：把两个条件写在同一行上

Student	Sno	Sname	Ssex	Sage	Sdept
	P.200215121			>19	CS

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	Sage	Sdept
	P.200215121 P.200215121			>19	CS

条件查询（与条件）

[例5] 查询既选修了1号课程又选修了2号课程的学生的学号。

Sc	Sno	Cno	Grade
	P.200215121	1	
	P.200215121	2	

[例6]查询计算机科学系或者年龄大于19岁的学生的学号。

Student	Sno	Sname	Ssex	Sage	Sdept
	P.200215121				CS
	P.200215122			>19	

条件查询（多表连接）

[例7] 查询选修1号课程的学生姓名。

Student	Sno	Sname	Ssex	Sage	Sdept
	200215121	P.李勇			

Sc	Sno	Cno	Grade
	200215121	1	

注意：示例元素Sno是连接属性，其值在两个表中要相同。

条件查询（非条件）

[例8] 查询未选修1号课程的学生姓名

Student	Sno	Sname	Ssex	Sage	Sdept
	200215121	P.李勇			

Sc	Sno	Cno	Grade
¬	200215121	1	

思路：显示学号为200215121的学生名字，而该学生选修1号课程的情况为假

条件查询（续）

[例9] 查询有两个人以上选修的课程号。

Sc	Sno	Cno	Grade
	200215121 ¬ 200215121	P.1 1	

思路：查询这样的课程1，它不仅被200215121选修
而且也被另一个学生（¬200215121）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

[例10] 查询信息系学生的平均年龄。

Student	Sno	Sname	Ssex	Sage	Sdept
				P.AVG.ALL	IS

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11] 查全体男生的姓名，要求查询结果按所在系升序排序，对相同系的学生按年龄降序排序。

Student	Sno	Sname	Ssex	Sage	Sdept
		P.李勇	男	DO（2）.	AO（1）.

二、更新操作

1.修改操作

[例12] 把200215121学生的年龄改为18岁。

方法(1)：将操作符“U.”放在值上

Student	Sno	Sname	Ssex	Sage	Sdept
	200215121			U.18	

方法(2)： 将操作符“U.”放在关系上

Student	Sno	Sname	Ssex	Sage	Sdept
U.	200215121			18	

码200215121标明要修改的元组。

“U.”标明所在的行是修改后的新值。

由于主码是不能修改的，**所以系统不会混淆要修改的属性。**

修改操作(续)

[例13] 把200215121学生的年龄增加1岁

Student	Sno	Sname	Ssex	Sage	Sdept
	200215121			17	
U.	200215121			17+1	

操作涉及表达式，必须将操作符“U.”放在关系上

[例14] 将计算机系所有学生的年龄都增加1岁

Student	Sno	Sname	Ssex	Sage	Sdept
	200215122			18	CS
U.	200215122			18+1	

2.插入操作

[例15] 把信息系女生200215701，姓名张三，年龄17岁存入数据库中。

Student	Sno	Sname	Ssex	Sage	Sdept
I.	200215701	张三	女	17	IS

3. 删除操作

[例17] 删除学生200215089

Student	Sno	Sname	Ssex	Sage	Sdept
D.	200215089				

为保证参照完整性，删除200215089学生前，先删除200215089学生选修的全部课程

Sc	Sno	Cno	Grade
D.	200215089		

第二章 关系数据库

2.6 小结

- 关系数据库系统是目前使用最广泛的数据库系统
- 关系数据库系统与非关系数据库系统的区别：
  - 关系系统只有“表”这一种数据结构；
  - 非关系数据库系统还有其他数据结构，以及对这些数据结构的操作

小结（续）

- 关系数据结构
  - 关系
    - 域
    - 笛卡尔积
    - 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

小结（续）

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

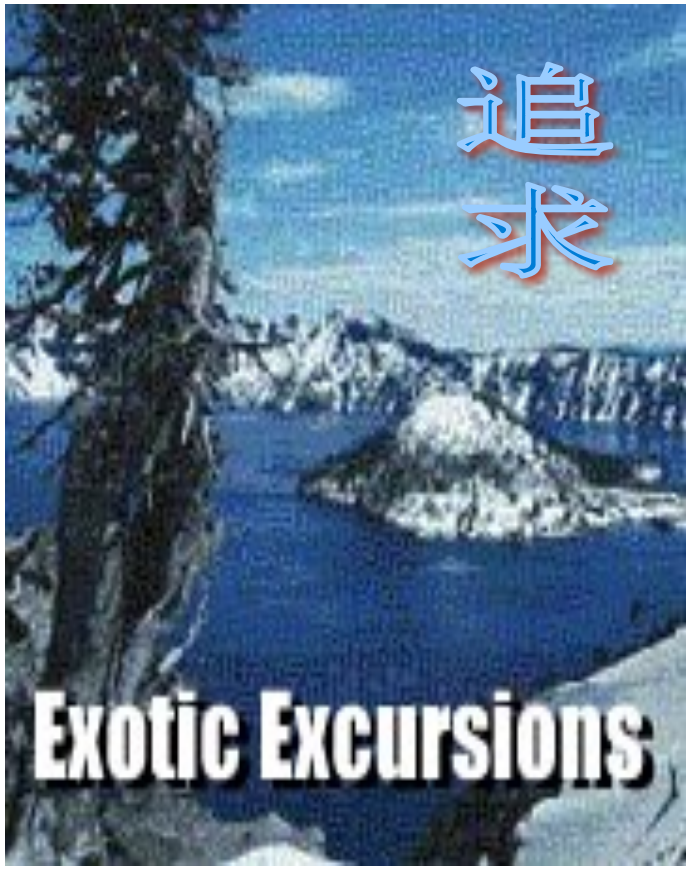
小结（续）

- 关系的完整性约束
  - 实体完整性
  - 参照完整性
    - 外码
  - 用户定义的完整性

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

下课了。。。



休息一会儿。。。