第2章 关系数据库

本章目标

- 完成本章的学习,你应该能够

- 理解关系模型的三要素
- 理解并区分关系模型的术语
- 掌握关系数据库的数学定义
- 了解关系的三种类型
- 理解基本关系的六个性质
- 区分关系模式和关系
- 理解实体完整性、参照完整性和自定义的完整性的含义
- 如何确定候选码、主码、外码
- 利用关系代数实现增删改查功能

大纲

- 关系数据结构及形式化定义
- 关系操作
- 关系的完整性
- 关系代数
- 本章小结

关系数据库简介

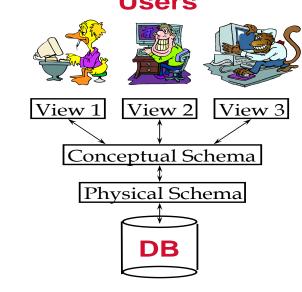
- 提出关系模型的是美国IBM公司的E.F.Codd

- 1970, E.F.Codd, "A Relational Model of Data for Large Shared Data Banks", 《Communication of the ACM》
 - 1983年ACM把该论文列为从1958年以来的四分之一世纪中具有里程碑意义的25篇研究论文之一。
- 1972, 提出关系代数和关系演算, 第一、第二、第三范式
- 1974, 提出关系的BC范式
- 1990, 出版专著《 The Relational Model for Database Management: Version 2 》
- 1981, 获图灵奖, 演讲题目 "Relational Database: A Practical Foundation for Productivity"

关系模型概述

关系数据库系统是支持<u>关系模型</u>的数据库系统

- 关系模型的组成
 - 关系数据结构+关系操作集合+关系完整性约束
- 关系数据结构
 - 单一数据结构:关系(二维表)
 - 简单但语义表达丰富: 现实世界的实体以及实体间的各种联系均用关系来表示
 - 建立在集合代数的基础上



关系数据结构及形式化定义

- 关系(Relation)
- 关系模式(Relation Schema)
- 关系数据库(Relational Database)
- 关系模型的存储结构(Relational Model Storage)

关系

- 域(Domain)

- 一组具有相同数据类型的值的集合,如,整数,实数,{'男','女'},字符串,日期,...

Attribute	Domain Name	Meaning	Domain Definition
branchNo street city postcode sex DOB	BranchNumbers StreetNames CityNames Postcodes Sex DatesOfBirth Salaries	The set of all possible branch numbers The set of all street names in Britain The set of all city names in Britain The set of all postcodes in Britain The sex of a person Possible values of staff birth dates Possible values of staff salaries	character: size 4, range B001–B999 character: size 25 character: size 15 character: size 8 character: size 1, value M or F date, range from 1-Jan-20, format dd-mmm-yy monetary: 7 digits, range 6000.00–40000.00

笛卡尔积(Cartesian Product)

- 给定一组域D₁, D₂, ..., D_n, <u>允许其中某些域相同</u>, D₁, D₂, ..., D_n的笛卡尔积为:
 D₁×D₂×...×D_n = {(d₁, d₂, ..., d_n) | d_i ∈ D_i, i = 1, 2, ..., n}
 - 其中,每个元素(d₁, d₂, ..., d_n)称为一个n元组(n-tuple),元素中的每个值d_i 称为一个分量 (component)
 - 是所有域所有取值的一个组合
 - 不能重复
- 基数(Cardinality): $|D_1| \times |D_2| \times ... \times |D_n|$
- 可表示为一个二维表, 表中行对应元组, 列对应域

```
D_1 = \{1, 3\} \quad D_2 = \{2, 4\} \quad D_3 = \{5, 6\}
D_1 \times D_2 \times D_3 = \{(1, 2, 5), (1, 2, 6), (1, 4, 5), (1, 4, 6), (3, 2, 5), (3, 2, 6), (3, 4, 5), (3, 4, 6)\}
```

关系(Relation):

- D₁×D₂×...×D_n的子集叫做域D₁,D₂, ...,D_n上的关系,记为R(D₁, D₂, ..., D_n)
 - R: 关系名
 - n: 关系的目或度(degree)
 - 关系中的每个元素是关系中的元组(tuple),通常用t表示
 - n=1: 一元关系或单元关系(Unary relation)
 - n=2: 二元关系(Binary relation)

• 关系的表示:

- 可视为一张二维表, 表的每行对应一个元组, 表的每列对应一个域

关系是笛卡尔积的有限子集。无限关系在数据库系统中是无意义的。由于笛卡尔积不满足交换律,即

$$(d_{1,}d_{2,}...,d_{n}) \neq (d_{2,}d_{1,}...,d_{n})$$

但关系满足交换律,即

$$(d_1,d_2,...,d_i,d_j,...,d_n)=(d_1,d_2,...,d_i,d_i,...,d_n)(i,j=1,2,...,n)$$

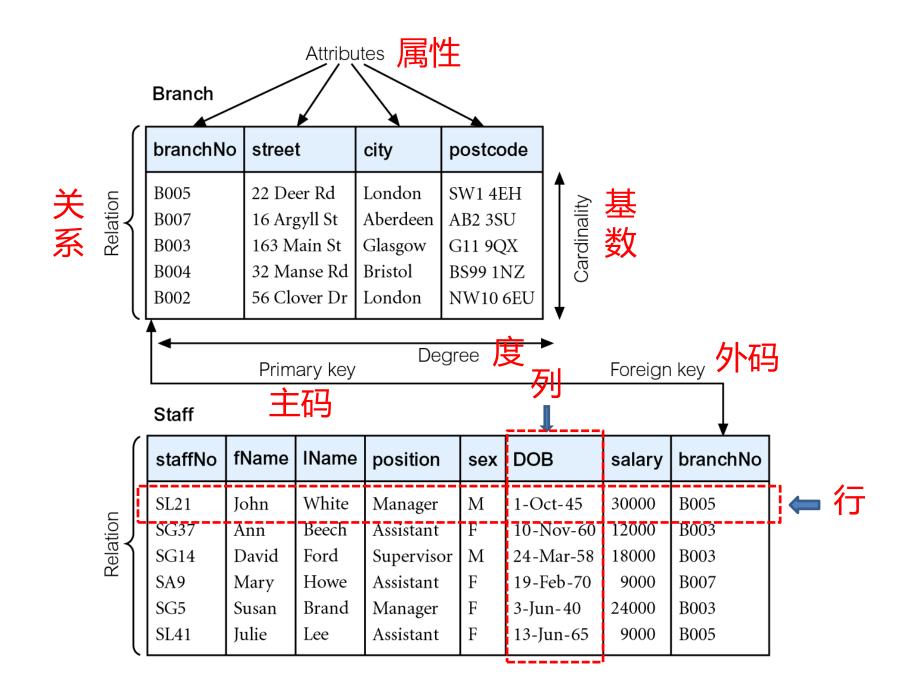
- 解决方法:
 - 为关系的每列附加一个属性名以取消关系元组的有序性

- 属性(Attribute)

- 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
- n目关系必有n个属性

• 码(key)

- 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,则称该属性组为该关系的一个候选码
- 主码(Primary Key, PK): 若一个关系有多个候选码,则选定其中一个为主码
- 全码(All key): 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
- 主属性(primary attribute): 候选码的所有属性
- 非主属性(非码属性): 不包含在任何候选码的属性



• 关系的三种类型:

- 基本关系(基本表或基表, Base Table)
 - 实际存在的表,是实际存储数据的逻辑表示
- 查询表(Query table)
 - 查询结果对应的表
- 视图(View)
 - 由基表或其他视图表导出,是虚表,不对应实际存储的数据
 - •注:物化视图是唯一例外,它存储实际数据,主要用于缓存复杂查询的结果,可周期性<mark>刷新</mark> (refresh)
 - CREATE MATERIALIZED VIEW view_table, ...

• 基本关系的六个性质:

- 三列两行一分量
 - 列是同质的(Homogeneous)
 - 不同的列可出自同一个域,每一列为一个属性,不同属性(列)给予不同属性名
 - 列的次序可以任意交换
 - 任意两个元组的候选码不能相同
 - 行的次序可任意交换
 - 分量必须取原子值,即每一分量是不可分的数据项



在许多实际关系数据库产品中,基本表并不完全具有这六条性质,如,FoxPro仍然区分了属性顺序和元组的顺序;Oracle允许关系表中存在两个完全相同的元组

• 范式(NF)

- 规范化的关系简称范式(Normal Form, NF)
 - 第一范式, 第二范式, 第三范式, 第四范式, BC范式, 第五范式
- 关系模式要求关系必须是规范化(Normalization)的,即要求关系必须满足一定的规范条件
 - 性质6是最基本的一条,即数据项是不可分的原子值

SUPERVISOR	SPECIALITY	POSTGRADUATE 6		
	SI ECIALITI	PG1	PG2	
张清玫	信息专业	李勇	刘晨	
刘逸	信息专业	王敏		小表

关系数据结构及形式化定义

- 关系(Relation)
- 关系模式(Relation Schema)
- 关系数据库(Relational Database)
- 关系模型的存储结构(Relational Model Storage)

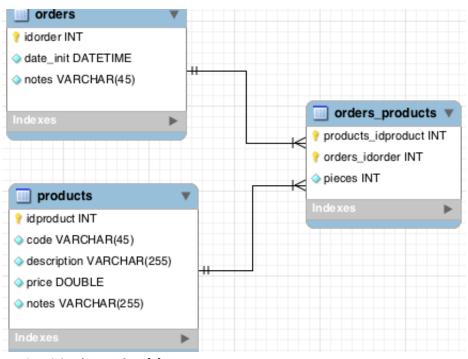
关系模式

• 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

- 完整性约束条件

- 约束或通过对属性取值范围的限定
 - 如成绩介于85-100
- 通过属性值间的相互关联反映出来(如2个元组的主码相等)



• 关系模式的形式化定义:

- 五元组R(U, D, DOM, F)
 - R: 关系名
 - U: 关系的属性名集合
 - D: U中属性所来自的DOM属性向域的映象集合
 - F: 属性间的数据依赖关系集合
- 简记为R(U), 或R(A1,A2,...,An), Ai为属性名
 - 域名及属性向域的映象常常直接说明为属性的类型、长度

```
CREATE TABLE orders(
idorder INT PRIMARY KEY,
date_init DATETIME,
notes VARCHAR(45));
```

关系模式与关系的联系与区别

关系模式	关系
• 静态	• 动态
• 稳定	• 随时间变化
• 型	• 值
_\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	

• 关系模式与关系往往笼统称为关系,但可以通过上下文加以区别

问题:在实际操作过程中,是先创建关系模式再创建关系,还是相反?



S001	张三	21
S002	李四	20
ສ003	王五	22

C001	数据库	4
C002	英语	б
C003	数学	б

S001	C001	90
S002	C001	80
S003	C001	90
S003	C002	96
S003	C003	98

学生(学号,姓名,年龄) 课程(<u>课程号</u>,课程名,学分) 选课(学号,<u>课程号</u>,成绩)

关系数据结构及形式化定义

- 关系(Relation)
- 关系模式(Relation Schema)
- 关系数据库(Relational Database)
- 关系模型的存储结构(Relational Model Storage)

关系数据库

• 关系数据库

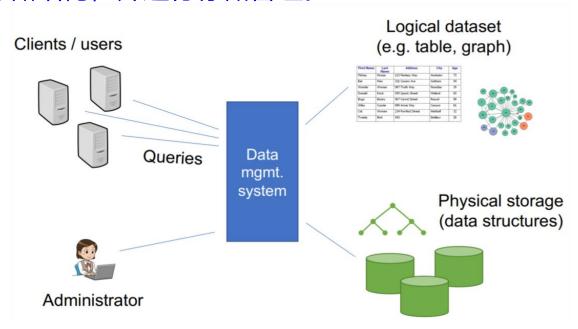
- 在一个给定的应用领域中,所有关系的集合构成一个关系数据库
- 关系数据库的型(Type)与值(Value)
 - 关系数据库的型(Type)
 - 即关系数据库模式 R(U)
 - 是对关系数据库的描述,包括若干域的定义以及在这些域上定义的若干关系模式
 - 可使用SQL语句 CREATE TABLE, ALTER TABLE命令实现
 - 关系数据库的值(Value)
 - 关系模式在某一时刻对应的关系的集合,通常简称关系数据库

关系数据结构及形式化定义

- 关系(Relation)
- 关系模式(Relation Schema)
- 关系数据库(Relational Database)
- 关系模型的存储结构(Relational Model Storage)

关系模型的存储结构

- 关系模型的存储结构指的是关系数据库的物理组织,有时也称 为关系数据库的物理结构
 - 具体实现由关系数据库管理系统产品决定:一个表对应一个操作系统文件,将物理数据组织交给操作系统完成;或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。



大纲

- 关系数据结构及形式化定义
- 关系操作
- 关系的完整性
- 关系代数
- 本章小结

关系操作

- 关系模型给出了关系操作能力的说明,但不对关系数据库管理系统 语言给出具体的语法要求,即不同的关系数据库管理系统可以定义 和开发不同的语言来实现这些操作。
- 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
 - 数据更新: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, 一次一集合的方式

• 关系数据库语言分类

关系代数语言	关系演算语言	具有关系代数和关系演算 双重特点的语言
• 用关系代数表达查询要求	• 用谓词来表达查询要求	・代表: SQL
• 代表: ISBL	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	

关系代数、元组关系演算、域关系演算、SQL的表达能力等价

大纲

- 关系数据结构及形式化定义
- 关系操作
- 关系的完整性
- 关系代数
- 本章小结

关系的三类完整性约束

- 关系模型的完整性规则是对关系的某种约束条件
 - 这些约束是现实世界的要求。任何关系在任何时刻都要满足这些语义约束
- 关系模型中三类完整性约束(constraint)
 - 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

实体完整性

• 实体完整性规则

- 若属性A是基本关系R的主属性,则属性A不能取空值
- 空值(Null)就是"不知道"或"不存在"或"无意义"的值

- 示例:

- 学生(学号,姓名,性别,专业号,年龄),"学号"为主码,不能取空值
- 选修(学号, 课程号,成绩)
- (学号, 课程号)为主码, 这两个属性都不能取空值
 - 学号为主属性; 课程号为主属性

- 实体完整性规则说明

- 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
- 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
 - 关系模型中以主码作为唯一性标识
- 主码中的属性即主属性不能取空值
 - 如果主属性取空值,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
 - 问题: 主属性是指候选码的属性, 那么如何保证非主码的主属性不空?
 - 方法: 在定义表的时候要求该属性非空即可

参照完整性

• 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

■ 示例:

```
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
801	张三	女	01	19
802	李四	男	01	20
803	王五	男	01	20
804	赵六	女	02	20
805	钱七	男	02	19

专业(专业号,专业名)

专业号	专业名
01	信息
02	数学
03	计算机

学生(学号,姓名,性别,专业号,年龄) 选修(学号,课程号,成绩) 课程(课程号,课程名,学分)

	/学号		/ 课程号	成绩
	801	\ /	04	92
1	801	۱ /	03	78
1	801		02	85
١	802	/ \	03	82
1	802	\	04	90
	803		04	88
	\ /			

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

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(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

外码(Foreign Key, FK)

- 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Ks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
- 基本关系S称为被参照关系或目标关系

说明:

- R和S不一定是不同的关系;
- S的主码K。与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

• 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

■ 示例:

- 学生关系中每个元组的"专业号"属性只取两类值:空值或非空值;
- 选修(学号, 课程号, 成绩)中"学号", "课程号"的非空取值;
- 学生(学号, 姓名, 性别, 专业号, 年龄, 班长)中"班长"的空值或非空取值

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
- 用户定义的完整性主要体现在:
 - 数据类型,取值范围,能否取空值
- 示例:
 - 课程(课程号,课程名,学分)
 - 课程号取唯一值
 - 课程名非空
 - 学分取{1, 2, 3, 4, 5}

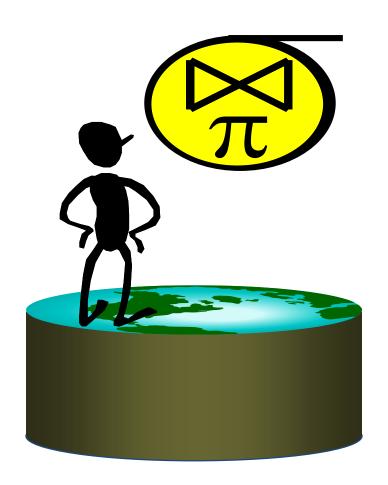
```
-- Drop student table, and then create it again.
DROP TABLE Student;
CREATE TABLE Student
     (Sno CHAR(9) PRIMARY KEY, /* 列级完整性约束条件, Sno是主码*/
      Sname CHAR(20) UNIQUE, --Sname取唯一值
           CHAR (2),
      Ssex
      Sage SMALLINT,
      Sdept CHAR (20)
     );
--Drop course table, and then create it again.
DROP TABLE Course;
CREATE TABLE Course
            (Cno CHAR (4) PRIMARY KEY,
            Cname CHAR (40),
             Cpno CHAR (4),
            Ccredit SMALLINT,
             FOREIGN KEY (Cpno) REFERENCES Course (Cno)
            );
--Deal with SC table
DROP TABLE SC;
CREATE TABLE SC
         (Sno CHAR(9),
          Cno CHAR(4),
          Grade SMALLINT,
          PRIMARY KEY (Sno, Cno),
          FOREIGN KEY (Sno) REFERENCES Student (Sno),
          /* 表级完整性约束条件, Sno是外码, 被参照表是Student */
          FOREIGN KEY (Cno) REFERENCES Course (Cno)
          /* 表级完整性约束条件, Cno是外码,被参照表是Course*/
       );
```

SQL语句完成:

- 创建表
- 定义完整性

大纲

- 关系数据结构及形式化定义
- 关系操作
- 关系的完整性
- 关系代数
- 本章小结



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
 - 关系代数的运算符有两类: 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

• 关系代数运算符:

运	算符	含义	运	算符	含义
集合 运算 符	U - n x	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

•并、差、交、 广义笛卡尔积

•选择、投影、连接、除

运算符	含义		运算符	含义	
专门的关 系运算符	σ π X ÷	选择 投影 连接	逻辑运算 符	「<>	非与或

■ 传统的集合运算: 并(Union)

 $- R \cup S = \{ t \mid t \in R \lor t \in S \}$

_	R				$R \cup S$		
_	A	В	C		,		~
	a_1	b_1	c_1		A	В	C
	a_1	b_2	c_2		a_1	b_1	c_1
_	a_2	b_2	c_1		•		
	S			_ 7	a_1	b_2	c_2
	A	В	C	_	a_2	b_2	c_1
	a_1	b_2	c_2		a.	b_3	C.
	a_1	b_3	c_2		a_1	<i>U</i> ₃	c_2
	a_2	b_2	c_1				

■ 传统的集合运算: 差(Difference)

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

R		
A	В	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
S		
A	В	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

■ 传统的集合运算: 交(Intersection)

$$- R \cap S = \{ t \mid t \in R \land t \in S \} = R - (R-S)$$

R						
A	В	C		$D \cap C$		
a_1	b_1	c_1		$R \cap S$		
a_1	b_2	c_2		A	B	C
a_2	b_2	c_1				
S			_ 7	a_1	b_2	c_2
A	В	C	_	a	, , , , , , , , , , , , , , , , , , ,	
a_1	b_2	c_2		a_2	b_2	c_1
a_1	b_3	c_2				
a_2	b_2	c_1				

■ 传统的集合运算: 笛卡尔积(Cartesian Product)

$$-R \times S = \{ t_r t_s \mid t_r \in R \land t_s \in S \}$$

_	10			
	A	В	C	
	a_1	b_1	c_1	
	a_1	b_2	c_2	
	a_2	b_2	c_1	
	S			
	A	В	C	
	a_1	b_2	c_2	
	a_1	b_3	c_2	
	a_2	b_2	c_1	



 $R \times S$

K × 5							
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		

• 专门的关系运算

- -R, $t \in R$, $t[A_i]$
 - 设关系模式为R(A1, A2, ..., An)
 - 它的一个关系设为R
 - t∈R表示t是R的一个元组
 - t[Ai]则表示元组t中相应于属性Ai的一个分量
- -A, t[A], \overline{A}
 - 若 $A=\{A_{i_1},A_{i_2},...,A_{i_k}\}$,其中 $A_{i_1},A_{i_2},...,A_{i_k}$ 是 $A_1,A_2,...,A_n$ 中的一部分,则A称为<u>属性列或属性组</u>
 - $t[A]=(t[A_{i_1}], t[A_{i_2}], ..., t[A_{i_\ell}])$ 表示元组t在属性列A上诸分量的集合
 - \overline{A} 则表示 $\{A_1, A_2, ..., A_n\}$ 中去掉 $\{A_{i_1}, A_{i_2}, ..., A_{i_k}\}$ 后剩余的属性组
- $-t_{\rm r}t_{\rm s}$
 - R为n目关系, S为m目关系, $t_r \in \mathbb{R}$, $t_s \in \mathbb{S}$, $t_r t_s$ 称为元组的连接

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:

$$\mathbf{Z}_{x} = \{t[Z] | t \in \mathbb{R}, t[X] = x\}$$

• 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

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

• x₁在R中的象集

$$Z_{x_1} = \{Z_1, Z_2, Z_3\},\$$

• x₂在R中的象集

$$Z_{x_2} = \{Z_2, Z_3\}$$
,

• x₃在R中的象集

$$Z_{x_3} = \{Z_1, Z_3\}$$

■ 四种专门的关系运算:

- 选择(Selection)
- 投影(Projection)
- 连接(Join)
- 除(Division)

学生-课程数据库

- 数据库由三张表组成:
 - 学生关系Student
 - 课程关系Course
 - 课程选修关系SC

SC

学号Sno	课程号Cno	成绩Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

Student

学号Sno 年龄 所在系 姓名 性别 **Sdept Sname** Ssex Sage 男 201215121 李勇 **20** CS 刘晨 女 201215122 CS **19** 王敏 女 201215123 **18** MA 男 张立 201215125 **19** IS

Course

课程号Cno	课程名Cname	先行课pno	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

选择(Selection):

- 选择又称为限制(Restriction)
- $-\sigma_{F}(R) = \{t \mid t \in R \land F(t) = '\rlap{\ }\underline{a}'\}$
- 在关系R中选择满足给定条件的诸元组
- F: 选择条件,逻辑表达式,取值为"真"或"假"
 - ・基本形式为: X₁θY₁
 - X₁, Y₁等: 属性名、常量、简单函数; 属性名也可以用它的序号来代替;
 - θ表示比较运算符,它可以是>,≥,<,≤,=或<>
- 对应于SQL语句中的WHERE子句

• 选择示例:

- 查询信息系(IS系)全体学生

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

Sno	Sname	Ssex	Sage	Sdept
201215125	张立	男	19	IS

- 查询年龄小于20岁的学生

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

Sno	Sname	Ssex	Sage	Sdept
201215122	刘晨	女	19	IS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

■ 投影(Projection):

- $-\pi_{A}(R) = \{ t[A] \mid t \in R \}$
- 从R 中选择出若干属性列组成新的关系
- A: R 中的属性列
- 对应于SQL语句中的SELECT后面的目标列,但目标列只能是R的属性,不能是其它表达式,包括别名
- 注意: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)

• 投影示例:

- 查询学生的姓名和所在系

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



Sname	Sdept
李勇	CS
刘晨	CS
王敏	MA
张立	IS

- 查询Student表中都有哪些系

• 连接(Jion):

- 也称为θ连接或内连接
- $R \bowtie_{A\theta B} S = \{ \widehat{t_r t_s} \mid t_r \in R \land t_s \in S \land t_r[A] \theta t_s[B] \}$
- 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
- 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
- 对应于SQL语句中的Join
- 连接运算从R和S的广义笛卡尔积R×S中选取(R关系)在A属性组上的值与(S关系)在B属性组上值满足比较关系θ的元组

• 两类常用连接:

- 等值连接(Equijoin)
 - θ 为 "=" 的连接运算称为等值连接

$$R \underset{A=B}{\bowtie} S = \{ \ \widehat{t_{\mathbf{r}}} \widehat{t_{\mathbf{s}}} \mid \underline{t_{\mathbf{r}}} \in \underline{R} \land \underline{t_{\mathbf{s}}} \in \underline{S} \land \underline{t_{\mathbf{r}}} [A] = \underline{t_{\mathbf{s}}} [B] \ \}$$

- 自然连接(Natural join)
 - 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

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

• 连接示例:

R		
A	В	С
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

В	E
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

 \mathcal{S}

一般连接 $R \bowtie S$ 的结果如下: $c < \epsilon$

 $R \bowtie S$ $C \le E$

C \L				
A	R.B	С	S.B	Ε
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

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

A	R.B	С	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

A	В	С	Е
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

- 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

◆ 外连接(➤, Outer join, Full outer join全外连接)

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 Value),就叫做外连接,是左外连接与右外连接的并集

左外连接(⋈, Left outer join, Left join)

- 只保留左边关系R中的悬浮元组就叫做左外连接

■ 右外连接(⋈, Right outer join, Right join)

- 只保留右边关系 R 中的悬浮元组就叫做左外连接

• 外连接示例:

A	В	C	Е
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

A	В	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

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

(b) 左外连接

 $R \bowtie S$

(c) 右外连接

 $R\bowtie S$

(a) 外连接

 $R \bowtie S$

• 问题:连接用于何种场景?

Employ	yee Table
LastName	DepartmentID
Rafferty	31
Jones	33
Steinberg	33
Robinson	34
Smith	34
Jasper	NULL

Department Table		
DepartmentID	DepartmentName	
31	Sales	
33	Engineering	
34	Clerical	
35	Marketing	

Example of an explicit inner join

SELECT *
FROM employee

INNER JOIN department

ON employee.DepartmentID = department.DepartmentID;

Example of an implicit inner join

SELECT *

FROM employee, department

WHERE employee.DepartmentID = department.DepartmentID;

Example of a left outer join

SELECT *

FROM employee LEFT OUTER JOIN department

ON employee.DepartmentID = department.DepartmentID;

通过连接运算可以实现多表查询

• Oracle的显式(隐式)内连接、外连接示例:

```
create table emp(
         empname varchar(20) not null,
         deptno char(10)
     insert into emp values('a','1');
     insert into emp values('b','2');
     insert into emp values('c','2');
     create table dept(
10
         deptno char(10) primary key,
11
         deptname varchar(20)
12
13
14
     insert into dept values('1','d');
15
16
     insert into dept values('2','e');
17
     insert into dept values('3','f');
```

```
SQL> select * from emp inner join dept on emp.deptno=dept.deptno;
EMPNAME
          DEPTNO
                     DEPTNO
                                DEPTNAME
                                d
SQL> select * from emp, dept where emp.deptno=dept.deptno;
EMPNAME
            DEPTNO
                        DEPTNO
                                    DEPTNAME
                                    d
SQL> select * from emp left outer join dept on emp.deptno=dept.deptno;
EMPNAME
         DEPTNO
                   DEPTNO
                             DEPTNAME
                             d
SQL> select * from emp natural join dept;
DEPTNO
             EMPNAME
                          DEPTNAME
                          d
             b
```

• 除(Division):

- 给定关系R (X, Y) 和S (Y, Z), 其中X, Y, Z为属性组
- 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 ÷S = $\{t_r[X] \mid t_r \in R \land \pi_Y(S) \subseteq Y_X\}$
 - Y_x: x在R 中的象集, x = t_r[X]

除示例:

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_3
 a_1
 b_2
 c_1

 (a)

S		
В	С	D
b_1	c_2	d_1
b_2	c_1	d_1
b_2	c_3	d_2

(b)

$$\begin{array}{c}
R \div S \\
\hline
A \\
\hline
a_1 \\
\hline
(c)
\end{array}$$

 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_3 的家集为{(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)}

• 除的实际应用:

- 设有一个现实意义的集合,希望在另一个集合中找出"包含"该集合的元组集,可用"除"实现

例1: 找出选修了所有课程的学生

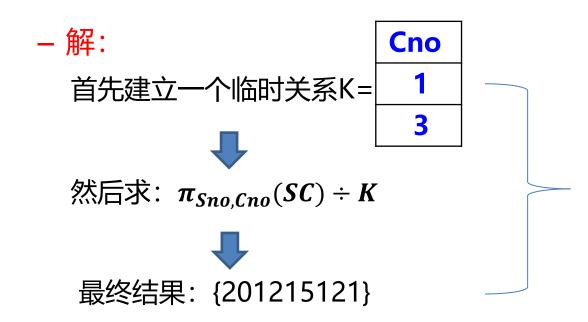
- "所有课程"
- "学生"
- "学生" : "所有课程"

例2: 找出选修了所有张三所选课的学生

- "张三所选课"
- "学生"
- "学生" ÷ "张三所选课"

除示例:

- 查询至少选修1号课程和3号课程的学生号码



Sno	Cno
201215121	1
201215121	2
201215121	3
201215122	2
201215122	3

综合示例:

- 查询选修了2号课程的学生学号

$$\pi_{sno}(\sigma_{cno='2'}(SC)) = \{201215121, 201215122\}$$

- 查询至少选修了一门其直接先修课为5号课程的学生姓名

$$\pi_{sname}\left(\sigma_{cpno='5'}\left(Course \bowtie SC \bowtie \pi_{sno, sname}(Student)\right)\right)$$

$$\pi_{sname}\left(\pi_{sno}\left(\sigma_{cpno='5'}(Course)\bowtie SC\right)\bowtie\pi_{sno,sname}(Student)\right)$$

- 查询选修了全部课程的学生的学号和姓名

$$\pi_{sno,cno}(SC) \div \pi_{Cno}(Course) \bowtie \pi_{sno,sname}(Student)$$

思考:上述查询方式哪种最差,哪个最优?

关系代数小结

• 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)

本章小结

- 介绍了关系模型的三个组成部分
 - 关系数据结构: 二维表
 - 关系操作: 关系代数
 - 完整性约束: 实体完整性、参照完整性和用户自定义完整性
- 关系代数是一种过程性语言(Procedural Language, PL),通过它可以实现关系数据库的增删改查。
 - 选择、投影、笛卡尔积、并和集合差为5种基本的操作
 - 连接、交和除则简化了操作

课堂练习

• 关于关系模型,下列叙述不正确的是().

A. 一个关系至少要有一个候选码 B. 列的次序可以任意交换

C. 行的次序可以任意交换

D. 一个列的值可以来自不同的域

• 关系操作中,操作的对象和结果都是().

A.记录

B.关系

C.元组

D.列

• 有两个关系R(A,B,C)和S(B,C,D),将R和S进行自然连接,得到的结果包含几个列().

A. 6

B. 4

C.5

D.6

- 判断题

判断以下论断是否正确。

- 1.关系模式是对关系的描述,关系是关系模式在某一时刻的状态或内容。()
- 2.关系模型的一个特点是,实体及实体间的联系都可以使用相同的结构类型来表示。()

• 填空题

- 1.在关系模型中,关系操作包括查询、_____、___和___等。
- 2.职工(<u>职工号</u>,姓名,年龄,部门号)和部门(<u>部门号</u>,部门名称)存在引用关系,其中是_____参照关系,____是外码。

本章作业

• 教材第二章习题: 4, 5, 8, 6(只要求用关系代数实现).