

第2章：关系数据库

Relational Databases

李东博

哈尔滨工业大学
计算学部
物联网与泛在智能研究中心
电子邮件: ldb@hit.edu.cn

2023年春

教学内容¹

① 关系数据模型

- ▶ 关系数据结构
- ▶ 关系操作
- ▶ 关系完整性约束

② 关系代数

- ▶ 基本关系代数操作
- ▶ 派生关系代数操作
- ▶ 扩展关系代数操作

③ 关系演算

- ▶ 元组关系演算
- ▶ 域关系演算

¹课件更新于2023年2月16日

2.1 关系数据模型

Relational Data Model

关系数据模型(Relational Data Model)

- 关系数据模型(relational data model)是一种被广泛使用的实现数据模型(implementation data model)
- 关系数据模型是关系数据库管理系统的模型基础
 - ▶ Oracle
 - ▶ Microsoft SQL Server
 - ▶ IBM DB2
 - ▶ MySQL
 - ▶ PostgreSQL
 - ▶ openGauss
 - ▶ SQLite

关系数据模型的三要素

- ① 关系数据结构
- ② 关系操作
- ③ 关系完整性约束

关系数据模型要素1: 关系数据结构

关系数据模型的三要素

- ① 关系数据结构
 - ② 关系操作
 - ③ 关系完整性约束
- 关系数据模型使用的唯一数据结构——关系(relation)
 - 不严格地讲, 关系就是一张二维表(table)
 - ▶ 行—元组(tuple)/记录(record), 表示对象
 - ▶ 列—属性(attribute)/域(field), 表示对象的性质

Example (关系)

Student				
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

关系(Relation)的数学定义 I

Definition (关系)

设 D_1, D_2, \dots, D_n 是 n 个值域(domain), $D_1 \times D_2 \times \dots \times D_n$ 的子集 R 称作 D_1, D_2, \dots, D_n 上的关系(relation), 记作 $R(D_1, D_2, \dots, D_n)$ 。

- R —关系名
- n —关系 R 的度(degree)
- $(d_1, d_2, \dots, d_n) \in R$ —关系 R 的元组(tuple), 其中 d_i 是元组的分量(component)

关系(Relation)的数学定义 II

Example (关系)

$D_1 =$ 学号集合, $D_2 =$ 姓名集合, $D_3 = \{M, F\}$, $D_4 = \mathbb{N}$, $D_5 =$ 系名集合

Student				
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

因为 $Student \subseteq D_1 \times D_2 \times D_3 \times D_4 \times D_5$, 所以 Student 是一个关系

关系的正确性

- $D_1 \times D_2 \times \dots \times D_n$ 的任意子集都是关系，但未必都是正确的关系
- 只有符合客观实际的关系才是正确的关系

Example (不正确的关系)

Student				
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math
MA-002	Cindy	F	20	Math

因为一个人不能同时有2个年龄，所以该Student关系是不正确的

关系的属性(Attributes)

Definition (属性)

由于域可能相同，为了加以区分，可为关系 $R(D_1, D_2, \dots, D_n)$ 的每个域 D_i 起一个不同的名字 A_i ，称作属性(attribute)，故关系 R 常表示为 $R(A_1, A_2, \dots, A_n)$ 。

Example (属性)

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

属性Sno的域是学号集合，属性Sname的域是姓名集合，属性Ssex的域是 $\{M, F\}$ ，属性Sage的域是 \mathbb{N} ，属性Sdept的域是系名集合

关系的键(Keys)

关系的某些属性集合具有区分不同元组的作用，称作键(key)

Definition (超键)

如果关系的某一组属性的值能唯一标识每个元组，则称该组属性为超键(super key)。

Example (超键)

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

属性集合{Sno, Cno}和{Sno, Cno, Grade}都是关系SC的超键

候选键(Candidate Keys)

Definition (候选键)

如果一个超键的任意真子集都不是超键，则称该超键为**候选键(candidate key)**。候选键=极小的(minimal)超键。

Example (候选键)

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

{Sno}和{Cno}都不是SC的超键，故{Sno, Cno}是SC的候选键

候选键是关系数据库规范化理论(第5章)中的重要概念!

主键(Primary Keys)

Definition (主键)

每个关系都有至少一个候选键，**人为指定**其中一个作为**主键(primary key)**。

Example (主键)

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

因为SC只有一个候选键，即{Sno, Cno}，所以SC的主键只能是{Sno, Cno} ▶ 演示

外键(Foreign Keys) I

不同关系中的元组可以存在联系，这种联系是通过外键建立起来的

Definition (外键)

设 F 是关系 R 的属性子集。若 F 与关系 S 的主键 K 相对应，则称 F 是 R 的**外键**(foreign key)

- R —参照关系(referring relation)
- S —被参照关系(referred relation)
- R 与 S 可以是同一关系(什么情况下可以?)

外键(Foreign Keys) II

Example (外键)

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	

- SC.Sno和Student.Sno分别表示SC和Student的属性Sno
- SC.Sno是SC的外键，它参照Student.Sno ▶ 演示

关系数据模型要素2: 关系操作

关系数据模型的三要素

- ① 关系数据结构
 - ② 关系操作
 - ③ 关系完整性约束
- 查询操作: 从关系数据库中查找数据
 - 更新操作: 对关系数据库进行更新
 - ▶ 插入元组
 - ▶ 修改元组
 - ▶ 删除元组

查询语言(Query Languages)

查询语言(query language)是用于表示关系操作的语言

查询语言的类型

- 关系代数(relational algebra) (第2.2节)
 - ▶ 使用关系代数表达式明确给出查询的执行过程
- 关系演算(relational calculus) (第2.3节)
 - ▶ 使用谓词逻辑表达式描述查询
 - ▶ 元组关系演算(tuple relational calculus): 谓词逻辑变量是元组
 - ▶ 域关系演算(domain relational calculus): 谓词逻辑变量是域
- 结构化查询语言SQL (第3章)
 - ▶ 具有关系代数和关系演算的双重特点
 - ▶ 集DDL、DML、DCL于一体

关系数据模型要素3: 关系完整性约束

关系数据模型的三要素

- ① 关系数据结构
- ② 关系操作
- ③ 关系完整性约束

- 完整性约束(integrity constraints): 关系数据库中的所有数据必须满足的约束条件
- 完整性约束的类型
 - ① 实体完整性(entity integrity)
 - ② 参照完整性(referential integrity)
 - ③ 用户定义完整性(user-defined integrity)

实体完整性约束 I

实体完整性约束规则

- ① 关系中任意元组的主键值必须**唯一(unique)**
- ② 关系中任意元组在主键中的属性值**非空(not null)**
 - ▶ 空值(null)表示值不存在，它既不是0，也不是空串

实体完整性约束 II

Example (实体完整性约束)

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

- {Sno, Cno} 是 SC 的主键
- 所有元组的 Sno 和 Cno 属性值组合必须唯一 ▶ 演示
- 任意元组的 Sno 和 Cno 属性值必须非空 ▶ 演示

参照完整性约束 I

不同关系中的元组可以存在联系，这种联系是通过外键建立起来的

参照完整性约束规则

设 F 是关系 R 的外键， F 参照关系 S 的主键，则 R 中任意元组的 F 属性值必须满足以下两个条件之一：

- ① F 的值为空
- ② 若 F 的值不为空，则 F 的值必须在 S 中存在

参照完整性约束 II

Example (参照完整性约束)

SC

Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

Student

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	

- SC.Sno和Student.Sno分别表示SC和Student的属性Sno
- SC.Sno是SC的外键，它参照Student.Sno
- SC.Sno的属性值集合必须是Student.Sno属性值集合的子集

▶ 演示

参照完整性约束 III

Example (参照完整性约束)

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	

Department	
Dept	Addr
Physics	B1
CS	B2
Math	B3

- Sdept是Student的外键，它参照Department的Dept属性
- Student中元组的Sdept属性值可以为空，表示该学生的院系未知
- 如果Student中元组的Sdept属性值非空，则该Sdept属性值必须属于Department中Dept的属性值集合

用户定义完整性约束

根据应用需求定义的完整性约束

Example (用户定义完整性约束)

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

- Student.Sname不可以为空
- Student.Ssex的值只能是'M'或'F'
- Student.Sage的值必须大于0

▶ 演示

关系的模式(Schema)

关系的模式(schema)是对关系的结构与语义的描述

- 关系名、属性名、属性值域、主键、完整性约束、属性依赖关系等
- 关系模式是不经常变化的

Example (关系模式)

Student				
Sno	Sname	Ssex	Sage	Sdept

查看关系模式

- PostgreSQL和openGauss: \d Student
- MySQL: describe Student

关系的实例(Instance)

关系的实例(instance)是关系在某一时刻的取值

- 关系实例必须符合关系模式
- 关系实例是动态变化的

关系模式与关系实例的关系如同面向对象程序设计中类(class)与对象(object)的关系

Example (关系实例)

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

查看关系实例: `SELECT * FROM Student;`

2.2 关系代数

Relational Algebra

关系代数(Relational Algebra)

- 关系代数是一种使用 **关系代数表达式** 来表示查询的语言
- 关系代数表达式明确给出了查询的执行过程
- 关系代数操作的 **操作数(operand)**: 关系
- 关系代数操作的 **结果**: 关系
- 关系代数操作的 **操作符(operator)**: 选择 σ 、投影 Π 、并 \cup 、差 $-$ 、笛卡尔积 \times 、重命名 ρ 、交 \cap 、内连接 \Join 、外连接 \Join 等

Example (关系代数表达式)

$\sigma_{Student.Sno=SC.Sno}(Student \times SC)$

- 关系 $Student$ 和 SC 是关系代数操作 \times (笛卡尔积)的操作数
- $Student \times SC$ 的结果也是关系, 它是关系代数操作 σ (选择)的操作数
- 上述关系代数表达式给出了执行过程: 先执行 \times 操作, 后执行 σ 操作

基本关系代数操作

基本关系代数操作

- ① 选择 σ
- ② 投影 Π
- ③ 并 \cup
- ④ 差 $-$
- ⑤ 笛卡尔积 \times
- ⑥ 重命名 ρ



电影《一代宗师》剧照

除一些特殊查询外，关系代数查询均可以由基本关系代数操作构成

选择操作(Selection)

- 功能: 从一个关系中选出满足给定条件的元组
- 语法: $\sigma_{\theta}(R)$
 - ▶ σ —选择操作符
 - ▶ R —关系名
 - ▶ θ —条件表达式, 形如 $A = 10$, $B > 5$ 的简单逻辑表达式, 或由与 \wedge 、或 \vee 、非 \neg 逻辑运算构成的复杂逻辑表达式

Example (选择操作)

① 找出计算机系的全体学生 $\sigma_{Sdept='CS'}(Student)$ ▶ 演示

② 找出计算机系的全体男同学 $\sigma_{Sdept='CS' \wedge Ssex='M'}(Student)$ ▶ 演示

Student

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

查询1的结果

Sno	Sname	Ssex	Sage	Sdept
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS

查询2的结果

Sno	Sname	Ssex	Sage	Sdept
CS-002	Ed	M	19	CS

投影操作(Projection)

- 功能: 从一个关系中选出指定的列, 并去掉重复元组
- 语法: $\Pi_L(R)$
 - ▶ Π —投影操作符
 - ▶ R —关系名
 - ▶ L —投影属性列表

Example (投影操作)

① 找出全体学生的学号和姓名 $\Pi_{Sno, Sname}(Student)$ ▶ 演示

② 找出全部的系 $\Pi_{Sdept}(Student)$ ▶ 演示

Student

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

查询1的结果

Sno	Sname
PH-001	Nick
CS-001	Elsa
CS-002	Eric
MA-001	Abby
MA-002	Cindy

查询2的结果

Sdept
Physics
CS
Math

并操作(Union)

- 功能: 计算关系 R 和 S 的并集
- 语法: $R \cup S$
 - ▶ R, S —关系名
 - ▶ \cup —并操作符
- 要求:
 - ① R 和 S 必须具有相同个数的属性
 - ② R 和 S 对应属性的值域必须相容

Example (集合并操作)

- ① 找出计算机系和数学系的学生

$\sigma_{Sdept='CS'}(Student) \cup \sigma_{Sdept='MA'}(Student)$ ▶ 演示 (还有什么方法?)

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

Sno	Sname	Ssex	Sage	Sdept
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS

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

Sno	Sname	Ssex	Sage	Sdept
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

查询1的结果

Sno	Sname	Ssex	Sage	Sdept
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

差操作(Difference)

- 功能: 计算关系 R 和 S 的差集
- 语法: $R - S$
 - ▶ R, S —关系名
 - ▶ $-$ —差操作符
- 要求:
 - ① R 和 S 必须具有相同个数的属性
 - ② R 和 S 对应属性的值域必须相容

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

Example (差操作)

- ① 查询选修了1002号课程, 但没有选修3006号课程的学生的学号

$\Pi_{Sno}(\sigma_{Cno='1002'}(SC)) - \Pi_{Sno}(\sigma_{Cno='3006'}(SC))$ ▶ 演示

$\Pi_{Sno}(\sigma_{Cno='1002'}(SC)) \quad \Pi_{Sno}(\sigma_{Cno='3006'}(SC))$

Sno
PH-001
CS-001
MA-001

-

Sno
PH-001
CS-001
CS-002

=

查询1的结果

Sno
MA-001

笛卡尔积操作 I

- 功能: 计算两个关系的笛卡尔积
- 语法: $R \times S$
 - ▶ R, S —关系名
 - ▶ \times —笛卡尔积操作符

笛卡尔积操作 II

Example (笛卡尔积操作)

Student

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

SC

Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

Student × *SC* ▶ 演示

Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
PH-001	Nick	M	20	Physics	PH-001	1002	92
PH-001	Nick	M	20	Physics	PH-001	2003	85
PH-001	Nick	M	20	Physics	PH-001	3006	88
PH-001	Nick	M	20	Physics	CS-001	1002	95
PH-001	Nick	M	20	Physics	CS-001	3006	90
PH-001	Nick	M	20	Physics	CS-002	3006	80
...

笛卡尔积操作

- 笛卡尔积的作用仅仅是将 R 和 S 中的元组无条件地连接起来
- 笛卡尔积操作通常和选择操作一起使用，即连接(join)

Example (笛卡尔积操作与选择操作结合)

① 查询已选课学生的信息

$\sigma_{Student.Sno=SC.Sno}(Student \times SC)$ ▶ 演示

查询1的结果

Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
PH-001	Nick	M	20	Physics	PH-001	1002	92
PH-001	Nick	M	20	Physics	PH-001	2003	85
PH-001	Nick	M	20	Physics	PH-001	3006	88
CS-001	Elsa	F	19	CS	CS-001	1002	95
CS-001	Elsa	F	19	CS	CS-001	3006	90
CS-002	Ed	M	19	CS	CS-002	3006	80
MA-001	Abby	F	18	Math	MA-001	1002	

重命名操作(Renaming)

- 功能: 修改关系名和(或)属性名
- 语法:
 - ① $\rho_{B \leftarrow A}(R)$: 将关系 R 的属性 A 更名为 B
 - ② $\rho_S(R)$: 将关系 R 更名为 S
 - ③ $\rho_{S(A_1, A_2, \dots, A_n)}(R)$: 将关系 R 更名为 S , 并将 R 的全部属性更名为 A_1, A_2, \dots, A_n
- 当把一个关系和它自身进行自连接(self-join)时, 需要区分同一个关系的两个副本。在这种情况下, 重命名操作发挥着重要作用。

Example (重命名操作)

- ① 将关系 SC 的属性名 Grade 修改为 Score

$\rho_{Score \leftarrow Grade}(SC)$ ▶ 演示

- ② 找出和 Elsa 在同一个系学习的学生的学号和姓名

$\Pi_{S2.Sno, S2.Sname}(\sigma_{S1.Sname='Elsa' \wedge S1.Sdept=S2.Sdept}(\rho_{S1}(Student) \times \rho_{S2}(Student)))$ ▶ 演示

- ③ ★★ 找出 3006 号课程的最高分(课后练习)

基本关系代数习题 I

使用关系代数运算器² 在数据库(Database Systems The Complete Book - Exercise 2.4.1)上完成下列习题

选择

- ① What PC models have a speed of at least 3.00 and ram of at least 1024MB?
- ② What PC models have a speed of at least 3.00 or ram of at least 1024MB?

投影

- ① What are the manufacturers?
- ② What models does the manufacturer A produce?
- ③ Find the model numbers of all color laser printers

并

- ① Find the model numbers and price of all PC's and all laptops

基本关系代数习题 II

差

- ① Find the manufacturers that sell laptops but not PC's

笛卡尔积

- ① What manufactures make laptops with a hard disk of at least 100GB?
- ② What PC models with a price less than \$500 does the manufacturer A produce?

重命名

- ① Rename the hd attribute of a PC to ssd
- ② ★★ Find the model numbers of all printers that are cheaper than the printer model 3002

²<https://dbis-uibk.github.io/relax>

派生关系代数操作

- 目的: 只用基本关系代数操作来编写复杂查询是非常繁琐的, 因此我们引入派生(derived)关系代数操作来简化查询编写
- 任何一项派生关系代数操作都可以用基本关系代数操作来表示

派生关系代数操作

- 1 交 \cap
- 2 θ 连接 \bowtie_{θ}
- 3 自然连接 \bowtie
- 4 外连接: 左外连接 $\bowtie\leftarrow$ 、右外连接 $\rightarrow\bowtie$ 、全外连接 $\bowtie\leftarrow\rightarrow$
- 5 反连接 \triangleright
- 6 除 \div

交操作(Intersection)

- 功能: 计算关系 R 和 S 的交集
- 语法: $R \cap S$
 - ▶ R, S —关系名
 - ▶ \cap —交操作符
- 要求:
 - ① R 和 S 必须具有相同个数的属性
 - ② R 和 S 对应属性的值域必须相容
- 等价变换: $R \cap S = R - (R - S)$

Example

- ① 查询既选修了1002号课程，又选修了3006号课程的学生的学号

$\Pi_{Sno}(\sigma_{Cno='1002'}(SC)) \cap \Pi_{Sno}(\sigma_{Cno='3006'}(SC))$ ▶ 演示

$\Pi_{Sno}(\sigma_{Cno='1002'}(SC)) \quad \Pi_{Sno}(\sigma_{Cno='3006'}(SC))$

Sno
PH-001
CS-001
MA-001

\cap

Sno
PH-001
CS-001
CS-002

$=$

查询1的结果

Sno
PH-001
CS-001

θ 连接(θ -Join) I

- 功能: 将关系 R 和 S 中满足给定连接条件 θ 的元组进行连接
- 语法: $R \bowtie_{\theta} S$
 - ▶ \bowtie —内连接操作符
 - ▶ θ —连接条件, 条件表达式的语法与选择操作条件相同
- $R \bowtie_{\theta} S$ 的结果包含 R 和 S 中的全部属性, 同名属性加关系名前缀

θ 连接(θ -Join) II

Example (θ 连接)

Student

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

SC

Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

$Student \bowtie_{Student.Sno=SC.Sno} SC$

演示

Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
PH-001	Nick	M	20	Physics	PH-001	1002	92
PH-001	Nick	M	20	Physics	PH-001	2003	85
PH-001	Nick	M	20	Physics	PH-001	3006	88
CS-001	Elsa	F	19	CS	CS-001	1002	95
CS-001	Elsa	F	19	CS	CS-001	3006	90
CS-002	Ed	M	19	CS	CS-002	3006	80
MA-001	Abby	F	18	Math	MA-001	1002	

θ 连接(θ -Join)

Property

$$R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$$

Example

- ① 查询计算机系学生的选课情况，列出学号、姓名、课号、得分
- $\Pi_{Student.Sno, Sname, Cno, Grade}(\sigma_{Sdept='CS'}(Student \bowtie_{Student.Sno=SC.Sno} SC))$
- ▶ 演示

$Student \bowtie_{Student.Sno=SC.Sno} SC$							
Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
PH-001	Nick	M	20	Physics	PH-001	1002	92
PH-001	Nick	M	20	Physics	PH-001	2003	85
PH-001	Nick	M	20	Physics	PH-001	3006	88
CS-001	Elsa	F	19	CS	CS-001	1002	95
CS-001	Elsa	F	19	CS	CS-001	3006	90
CS-002	Ed	M	19	CS	CS-002	3006	80
MA-001	Abby	F	18	Math	MA-001	1002	

等值连接(Equi-join)

等值连接(equi-join): 连接条件 θ 仅涉及相等比较的 θ 连接

自然连接(Natural Join) I

- 功能: 设 $\{A_1, A_2, \dots, A_k\}$ 是关系 R 和 S 的同名属性集合 $R.A_1 = S.A_1 \wedge R.A_2 = S.A_2 \wedge \dots \wedge R.A_k = S.A_k$
 - ① 从连接结果中去掉重复的同名属性(为什么?)
- 语法: $R \bowtie S$

自然连接(Natural Join) II

Example (自然连接)

Student

Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

SC

Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

Student ⋈ *SC* ▶ 演示

Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
PH-001	Nick	M	20	Physics	1002	92
PH-001	Nick	M	20	Physics	2003	85
PH-001	Nick	M	20	Physics	3006	88
CS-001	Elsa	F	19	CS	1002	95
CS-001	Elsa	F	19	CS	3006	90
CS-002	Ed	M	19	CS	3006	80
MA-001	Abby	F	18	Math	1002	

自然连接与 θ 连接的区别

	自然连接	θ 连接
连接条件	隐含给出	明确给出
连接结果的属性	去除重复的同名属性	保留重复的同名属性

Example (自然连接)

- ① 查询计算机系学生的选课情况，列出学号、姓名、课号、得分
 $\Pi_{Sno, Sname, Cno, Grade}(\sigma_{Sdept='CS'}(Student \bowtie SC))$

Student \bowtie SC						
Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
PH-001	Nick	M	20	Physics	1002	92
PH-001	Nick	M	20	Physics	2003	85
PH-001	Nick	M	20	Physics	3006	88
CS-001	Elsa	F	19	CS	1002	95
CS-001	Elsa	F	19	CS	3006	90
CS-002	Ed	M	19	CS	3006	80
MA-001	Abby	F	18	Math	1002	

内连接 (Inner Join)

- 前面讲的连接都属于内连接(inner join)
- R 和 S 的内连接($R \bowtie_{\theta} S$ 或 $R \bowtie S$)的结果只包含 R 和 S 中满足连接条件的元组
- R 和 S 中不满足连接条件的元组均不会出现在连接结果中

Example (内连接)

- 查询选过课的学生学号和姓名

$\Pi_{Sno, Sname}(Student \bowtie SC)$

▶ 演示

$Student \bowtie SC$						
Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
PH-001	Nick	M	20	Physics	1002	92
PH-001	Nick	M	20	Physics	2003	85
PH-001	Nick	M	20	Physics	3006	88
CS-001	Elsa	F	19	CS	1002	95
CS-001	Elsa	F	19	CS	3006	90
CS-002	Ed	M	19	CS	3006	80
MA-001	Abby	F	18	Math	1002	

因为 $Student \bowtie SC$ 是内连接，所以学生MA-002不会出现在连接结果中

外连接(Outer Join) I

- 除了要在 R 和 S 的连接结果中保留满足连接条件的全部元组外，在某些情况下，我们还需要在连接结果中保留 R 或(和) S 中的不满足连接条件的元组
- 在这种情况下，仅用内连接无法完成查询，因此引入外连接(outer join)

外连接(Outer Join) II

Example (外连接)

- ❶ 查询全体学生的选课情况(含未选课的学生)

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

想要的查询结果

Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
PH-001	Nick	M	20	Physics	PH-001	1002	92
PH-001	Nick	M	20	Physics	PH-001	2003	85
PH-001	Nick	M	20	Physics	PH-001	3006	88
CS-001	Elsa	F	19	CS	CS-001	1002	95
CS-001	Elsa	F	19	CS	CS-001	3006	90
CS-002	Ed	M	19	CS	CS-002	3006	80
MA-001	Abby	F	18	Math	MA-001	1002	
MA-002	Cindy	F	19	Math			

外连接的分类

- 左外连接(left outer join)
- 右外连接(right outer join)
- 全外连接(full outer join)

左外 θ 连接(Left Outer θ -Join) I

指定 R 为左关系(left relation), S 为右关系(right relation)

- 功能:

- ① 将 R 和 S 中满足给定连接条件 θ 的元组进行连接, 即计算 $R \bowtie_{\theta} S$
- ② 对于 R 中不满足给定连接条件 θ 的元组, 左外连接结果中也包含该元组, 只不过 S 中属性的值都为空(null)

- 语法: $R \Join_{\theta} S$

- ▶ \Join —左外连接操作符

左外 θ 连接(Left Outer θ -Join) II

Example (左外 θ 连接)

- ① 查询全体学生的选课情况(含未选课的学生)

$Student \bowtie_{Student.Sno=SC.Sno} SC$ ▶ 演示

$Student \bowtie_{Student.Sno=SC.Sno} SC$

Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
PH-001	Nick	M	20	Physics	PH-001	1002	92
PH-001	Nick	M	20	Physics	PH-001	2003	85
PH-001	Nick	M	20	Physics	PH-001	3006	88
CS-001	Elsa	F	19	CS	CS-001	1002	95
CS-001	Elsa	F	19	CS	CS-001	3006	90
CS-002	Ed	M	19	CS	CS-002	3006	80
MA-001	Abby	F	18	Math	MA-001	1002	
MA-002	Cindy	F	19	Math			

Question

为什么没有左内 θ 连接?

左外自然连接(Left Outer Natural Join) I

指定 R 为左关系(left relation), S 为右关系(right relation)

- 功能:

- ① 将 R 和 S 中满足自然连接条件的元组进行连接, 即计算 $R \bowtie S$
- ② 对于 R 中不满足自然连接条件 θ 的元组, 左外自然连接结果中也包含该元组, 只不过那些在 S 而不在 R 中的属性的值都为空(null)

- 语法: $R \Join S$

- ▶ \Join —左外连接操作符

左外自然连接(Left Outer Natural Join) II

Example (左外自然连接)

- ① 找出没选过课的同学的学号和姓名

$\Pi_{Sno, Sname}(\sigma_{Cno=NULL}(Student \bowtie SC))$ ▶ 演示

<i>Student</i> \bowtie <i>SC</i>						
Student.Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
PH-001	Nick	M	20	Physics	1002	92
PH-001	Nick	M	20	Physics	2003	85
PH-001	Nick	M	20	Physics	3006	88
CS-001	Elsa	F	19	CS	1002	95
CS-001	Elsa	F	19	CS	3006	90
CS-002	Ed	M	19	CS	3006	80
MA-001	Abby	F	18	Math	1002	
MA-002	Cindy	F	19			

右外连接(Right Outer Join)

指定 R 为左关系(left relation), S 为右关系(right relation)

右外 θ 连接

- 功能:

- ① 将 R 和 S 中满足给定连接条件 θ 的元组进行连接, 即计算 $R \bowtie_{\theta} S$
- ② 对于 S 中不满足给定连接条件 θ 的元组, 右外连接结果中也包含该元组, 只不过 R 中属性的值都为空(null)

- 语法: $R \bowtie_{\theta} S$

- ▶ \bowtie —右外连接操作符

右外自然连接

- 功能:

- ① 将 R 和 S 中满足自然连接条件的元组进行连接, 即计算 $R \bowtie S$
- ② 对于 S 中不满足自然连接条件的元组, 右外自然连接结果中也包含该元组, 只不过那些在 R 而不在 S 中的属性的值都为空(null)

- 语法: $R \bowtie_{\theta} S$

- ▶ \bowtie —右外连接操作符

全外连接(Full Outer Join)

指定 R 为左关系(left relation), S 为右关系(right relation)

全外 θ 连接

● 功能:

- ① 将 R 和 S 中满足给定连接条件 θ 的元组进行连接, 即计算 $R \bowtie_{\theta} S$
- ② 对于 R 中不满足给定连接条件 θ 的元组, 全外连接结果中也包含该元组, 只不过 S 中属性的值都为空(null)
- ③ 对于 S 中不满足给定连接条件 θ 的元组, 全外连接结果中也包含该元组, 只不过 R 中属性的值都为空(null)

● 语法: $R \bowtie_{\theta} S$

▶ \bowtie —全外连接操作符

全外自然连接: 定义留作练习

Property

- $R \bowtie_{\theta} S = (R \bowtie_{\theta} S) \cup (R \bowtie_{\theta} S)$
- $R \bowtie S = (R \bowtie S) \cup (R \bowtie S)$

反连接(Anti Join) I

指定 R 为左关系， S 为右关系

θ 反连接

- 功能:

- ① 找出 R 中不满足与 S 的连接条件 θ 的元组

- 语法: $R \triangleright_{\theta} S$

- ▶ \triangleright —反连接操作符

自然反连接

- 功能:

- ① 找出 R 中不满足与 S 的自然连接条件的元组

- 语法: $R \triangleright S$

反连接(Anti Join) II

Example (反连接)

- ① 找出没选过课的同学的学号和姓名 $\Pi_{Sno, Sname}(Student \triangleright SC)$ ▶ 演示

Student				
Sno	Sname	Ssex	Sage	Sdept
PH-001	Nick	M	20	Physics
CS-001	Elsa	F	19	CS
CS-002	Ed	M	19	CS
MA-001	Abby	F	18	Math
MA-002	Cindy	F	19	Math

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

Student \triangleright SC				
Sno	Sname	Ssex	Sage	Sdept
MA-002	Cindy	F	19	Math

除(Division)

- **目的**: 我们经常要做下面这种查询: 找出选修了**所有**课程的学生。
用基本关系代数操作来编写这种查询非常不便, 因此引入除操作。
- **整数除法**: 设 x 和 y 为正整数, $x \div y$ 的商是使得 $yz \leq x$ 的最大的整数 z
- **关系除法**
 - ▶ $R \div S$ 的结果是一个关系, 它只包含 R 中的属性, 但不包含 S 中的属性
 - ▶ $R \div S$ 的结果是**使得 $S \times T \subseteq R$ 的最大的关系 T**
- **语法**: $R \div S$
 - ▶ \div —除操作符

Sno	Cno
PH-001	1002
PH-001	2003
PH-001	3006
CS-001	1002
CS-001	3006
CS-002	3006
MA-001	1002

\div

Cno
1002
3006

$=$

Sno
PH-001
CS-001

除(Division)

Example (除)

- ① 找出选修了所有课程的学生学号

$\Pi_{Sno, Cno}(SC) \div \Pi_{Cno}(Course)$ ▶ 演示

$\Pi_{Sno, Cno}(SC)$

Sno	Cno
PH-001	1002
PH-001	2003
PH-001	3006
CS-001	1002
CS-001	3006
CS-002	3006
MA-001	1002

$\Pi_{Cno}(Course)$

Cno
1002
2003
3006

÷

=

查询1的结果

Sno
PH-001

派生关系代数习题 I

使用关系代数运算器³ 在数据库(Database Systems The Complete Book - Exercise 2.4.1)上完成下列习题

交

- ① Find the manufacturers that sell both laptops and PC's

θ 连接

- ① Find those pairs of PC models that have both the same speed and RAM. A pair should be listed only once
- ② \star Find those hard-disk sizes that occur in two or more PC's
- ③ $\star\star$ Find the PC model with the highest available speed
- ④ $\star\star$ Find the manufacturers of PC's with at least three different speeds

自然连接

- ① What manufacturers make laptops with a hard disk of at least 100GB?
- ② \star Explain the result of $Product \bowtie Printer$

派生关系代数习题 II

左外连接

- 1 Execute $Product \bowtie PC$
- 2 ** Find the PC model with the highest available speed (第2次出现, 上一次怎么做的?)

右外连接

- 1 Execute $Product \bowtie \lrcorner PC$

全外连接

- 1 Execute $Product \bowtie \ltimes PC$

除

- 1 What manufacturers make all types of products (PC, laptop, and printer)?

³<https://dbis-uibk.github.io/relax>

扩展关系代数操作

- **目的:** 用基本关系代数操作能够实现的查询功能有限, 为了增强关系代数的查询表示能力, 我们引入扩展(extended)关系代数操作

扩展关系代数操作

- ① 分组操作 γ
- ② 赋值操作 $=$

分组操作(Group-By)

- **目的**: 我们经常需要对数据进行统计, 例如统计每名学生的选课数和平均分。基本关系代数操作无法实现这种功能, 因此需要引入分组操作。
- **功能**:
 - ① 根据指定的分组属性, 对一个关系中的元组进行**分组**, 分组属性值相同元组的分为一组
 - ② 对每个组中元组的非分组属性的值进行**聚集(aggregation)**——计数count、求最小值min、求最大值max、求和sum、求平均值avg
 - ③ 聚集函数只作用于非空(null)值, count(*)除外(它计算分组内所有元组的数量)

SC		
Sno	Cno	Grade
PH-001	1002	92
PH-001	2003	85
PH-001	3006	88
CS-001	1002	95
CS-001	3006	90
CS-002	3006	80
MA-001	1002	

每名学生的选课数和平均分		
Sno	Amount	AvgGrade
PH-001	3	88.3
CS-001	2	92.5
CS-002	1	80
MA-001	1	

分组操作(Group-By)

● 语法: $\gamma_{L;agg}(R)$

- ▶ γ —分组操作符
- ▶ R —关系名
- ▶ L —分组属性列表, 用逗号分隔
- ▶ agg —聚集函数表达式列表, 用逗号分隔, 每个聚集函数表达式形如 $sum(score) \rightarrow TotalScore$ (计算score属性值的和, 并将结果命名为属性TotalScore)

Example (分组聚集)

① 统计每个系的男生人数和女生人数

$\gamma_{Sdept, Ssex; count(*)} \rightarrow Amt(Student)$ ▶ 演示

② 统计每名已选课学生的选课数和平均分

$\gamma_{Sno; count(*) \rightarrow Amt, avg(Grade) \rightarrow Score}(SC)$ ▶ 演示

SC		
Sdept	Ssex	Amt
Physics	M	1
CS	F	1
CS	M	1
Math	F	2

查询2的结果		
Sno	Amt	Score
PH-001	3	88.3
CS-001	2	92.5
CS-002	1	80
MA-001	1	

赋值操作(Assignment)

- **目的**: 仅用一个关系代数表达式来编写复杂查询通常会太冗长, 不易理解。为了便于理解, 需要将一个冗长的关系代数查询表达式分解为一系列简单的表达式, 这需要暂存一些中间结果。
- **功能**: 将关系代数查询表达式的结果赋值给临时关系
- **语法**: $R = expr$
 - ▶ R —临时关系名
 - ▶ $=$ —赋值操作符
 - ▶ $expr$ —关系代数查询表达式

扩展关系代数习题 I

使用关系代数运算器⁴ 在数据库(Database Systems The Complete Book - Exercise 2.4.1)上完成下列习题

分组

- ① How many models does every manufacturer have?
- ② How many models does every manufacturer have for every type of products?
- ③ ** Find those hard-disk sizes that occur in two or more PC's (第2次出现, 上一次是怎么做的?)
- ④ ** What manufacturers make all types of products (PC, laptop, and printer)? (第2次出现, 上一次是怎么做的?)

赋值

- ① ** What manufacturers make all types of products (PC, laptop, and printer)? (第3次出现, 以前两次是怎么做的?)

⁴<https://dbis-uibk.github.io/relax>

2.3 关系演算

Relational Calculus

元组关系演算(Tuple Relational Calculus)

元组关系演算(tuple relational calculus)用形如 $\{t|P(t)\}$ 的表达式表示查询

- t : 元组变量(tuple variable)
- P : 谓词(predicate)
- 元组关系演算表达式的结果是所有使谓词 P 为真的元组 t 的集合

记法

- $t[A]$: 元组 t 中属性 A 的值
- $t \in R$: t 是关系 R 中的元组
- \wedge : 合取
- \vee : 析取
- \neg : 否定
- \implies : 蕴含, $A \implies B \equiv \neg A \vee B$: “如果 A 为真, 则 B 为真”
- \forall : 全称量词, $\forall t(Q(t))$ 为真当且仅当任意元组 t 均使谓词 Q 为真
- \exists : 存在量词, $\exists t(Q(t))$ 为真当且仅当存在元组 t 使谓词 Q 为真

元组关系演算

Example (元组关系演算)

- ❶ 找出计算机系的全体学生

$$\{t | t \in Student \wedge t[Sdept] = 'CS'\}$$

- ❷ 找出计算机系和数学系的学生

$$\{t | t \in Student \wedge (t[Sdept] = 'CS' \vee t[Sdept] = 'MA')\}$$

- ❸ 找出全体学生的学号和姓名

$$\{t | \exists s \in Student (t[Sno] = s[Sno] \wedge t[Sname] = s[Sname])\}$$

- ❹ 查询既选修了1002号课程，又选修了3006号课程的学生学号

$$\{t | \exists s \in SC \exists s' \in SC (t[Sno] = s[Sno] = s'[Sno] \wedge s[Cno] = '1002' \wedge s'[Cno] = '3006'))\}$$

- ❺ 查询选修了1002号课程，但没有选修3006号课程的学生学号

$$\{t | \exists s \in SC (s[Sno] = t[Sno] \wedge s[Cno] = '1002') \wedge \forall s' \in SC (\neg (s'[Sno] = t[Sno] \wedge s'[Cno] = '3006'))\} \text{ 或}$$
$$\{t | \exists s \in SC (s[Sno] = t[Sno] \wedge s[Cno] = '1002') \wedge \forall s' \in SC (s'[Sno] = t[Sno] \implies s'[Cno] \neq '3006'))\}$$

元组关系演算

Example (元组关系演算)

- ① 查询已选课学生的学号和姓名

$$\{t \mid \exists s \in Student \exists r \in SC (t[Sno] = s[Sno] = r[Sno] \wedge t[Sname] = s[Sname])\}$$

- ② 找出没选过课的同学的学号和姓名

$$\{t \mid \exists s \in Student \forall r \in SC (t[Sno] = s[Sno] \wedge t[Sname] = s[Sname] \wedge s[Sno] \neq r[Sno])\}$$

- ③ 找出选修了所有课程的学生学号

$$\{t \mid \exists s \in SC \forall c \in Course (t[Sno] = s[Sno] \wedge s[Cno] = c[Cno])\}$$

- ④ 查询选修了CS-001号同学选修的所有课程的同学学号

$$\{t \mid \forall s \in SC (\exists s' \in SC (s'[Sno] = t[Sno] \wedge ((s[Sno] = 'CS-001') \implies (s'[Cno] = s[Cno])))\}$$

域关系演算(Domain Relational Calculus)

- 域关系演算(domain relational calculus)表达式与元组关系演算表达式的定义类似，不同之处是表达式中使用域变量(domain variable)，而不是元组变量
- 域关系演算表达式的一般形式为 $\{(x_1, x_2, \dots, x_n) | P(x_1, x_2, \dots, x_n)\}$
 - ▶ x_1, x_2, \dots, x_n : 域变量
 - ▶ P : 域关系演算公式
 - ▶ 域关系演算表达式的结果是所有使 $P(x_1, x_2, \dots, x_n)$ 为真的元组 (x_1, x_2, \dots, x_n) 的集合
- 记法
 - ▶ (x_1, x_2, \dots, x_n) : 域变量 x_1, x_2, \dots, x_n 构成的元组
 - ▶ $(x_1, x_2, \dots, x_n) \in R$: (x_1, x_2, \dots, x_n) 是关系 R 中的元组
 - ▶ \wedge : 合取
 - ▶ \vee : 析取
 - ▶ \neg : 否定
 - ▶ \implies : 蕴含
 - ▶ \forall : 全称量词
 - ▶ \exists : 存在量词

域关系演算

Example (域关系演算)

- ① 找出计算机系的全体学生

$$\{(n, m, s, a, d) | (n, m, s, a, d) \in Student \wedge d = 'CS'\}$$

- ② 找出计算机系和数学系的学生

$$\{(n, m, s, a, d) | (n, m, s, a, d) \in Student \wedge (d = 'CS' \vee d = 'MA')\}$$

- ③ 找出全体学生的学号和姓名

$$\{(n, m) | \exists s, a, d ((n, m, s, a, d) \in Student)\}$$

- ④ 查询既选修了1002号课程，又选修了3006号课程的学生的学号

$$\{(n) | \exists g ((n, '1002', g) \in SC) \wedge \exists g' ((n, '3006', g') \in SC)\}$$

- ⑤ 查询选修了1002号课程，但没有选修3006号课程的学生的学号

$$\{(n) | \exists g ((n, '1002', g) \in SC) \wedge \forall g' (\neg ((n, '3006', g') \in SC))\}$$

域关系演算

Example (域关系演算)

- ❶ 查询已选课学生的学号和姓名

$$\{(n, m) | \exists s, a, d((n, m, s, a, d) \in Student) \wedge \exists c, g((n, c, g) \in SC)\}$$

- ❷ 找出没选过课的同学的学号和姓名

$$\{(n, m) | \exists s, a, d((n, m, s, a, d) \in Student) \wedge \forall c, g(\neg((n, c, g) \in SC))\}$$

- ❸ 找出选修了所有课程的学生学号

$$\{n | \forall c(\exists t((c, t) \in Course) \wedge \exists g((n, c, g) \in SC))\}$$

- ❹ 查询选修了CS-001号同学选修的所有课程的同学学号

$$\{(n) | \forall c(\exists g, g'(((\text{'CS-001'}, c, g) \in SC) \implies ((n, c, g') \in SC)))\}$$

总结

① 关系数据模型

- ▶ 关系数据结构: 关系、属性、键
- ▶ 关系操作: 查询操作、更新操作(插入、修改、删除)、查询语言(关系代数、关系演算、SQL)
- ▶ 关系完整性约束: 实体完整性、参照完整性、用户定义完整性

② 关系代数

- ▶ 基本关系代数操作: 选择 σ 、投影 Π 、笛卡尔积 \times 、并 \cup 、差 $-$ 、重命名 ρ
- ▶ 派生关系代数操作: 交 \cap 、内连接 \bowtie_{θ} 、自然连接 \bowtie 、外连接(左外连接 $\bowtie\leftarrow$ 、右外连接 $\rightarrow\bowtie$ 、全外连接 $\bowtie\leftarrow\rightarrow$)、除 \div
- ▶ 扩展关系代数操作: 分组操作 γ 、赋值操作 $=$

③ 关系演算

- ▶ 元组关系演算
- ▶ 域关系演算

④ 在线练习: <https://dbis-uibk.github.io/relax>

① 用基本关系代数操作表示下列关系代数表达式

- ▶ $R \bowtie S$
- ▶ $R \div S$
- ▶ $R \Join S$

② 判断下列命题是否成立。若不成立，请给出反例。

- ① $\sigma_{\theta_1}(\sigma_{\theta_2}(R)) = \sigma_{\theta_2}(\sigma_{\theta_1}(R)) = \sigma_{\theta_1 \wedge \theta_2}(R)$
- ② $\Pi_{L_1}(\Pi_{L_2}(R)) = \Pi_{L_1}(R)$
- ③ $\Pi_L(\sigma_\theta(R)) = \sigma_\theta(\Pi_L(R))$
- ④ $\Pi_L(R \cup S) = \Pi_L(R) \cup \Pi_L(S)$
- ⑤ $\Pi_L(R \cap S) = \Pi_L(R) \cap \Pi_L(S)$
- ⑥ $\sigma_\theta(R \cap S) = \sigma_\theta(R) \cap S = R \cap \sigma_\theta(S)$
- ⑦ $\sigma_\theta(R - S) = \sigma_\theta(R) - S = R - \sigma_\theta(S)$
- ⑧ $(R \bowtie_{\theta_1} S) \bowtie_{\theta_2} T = R \bowtie_{\theta_1} (S \bowtie_{\theta_2} T)$
- ⑨ $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
- ⑩ $R \bowtie R = R \cap R$

习题 II

- ③ 设关系 $R(A, B)$ 中包含 $r > 0$ 个元组，关系 $S(B, C)$ 中包含 $s > 0$ 个元组，求下列关系代数表达式的结果中元组数的最小值和最大值

关系代数表达式	元组数最小值	元组数最大值
$\sigma_{A < B}(R)$		
$\Pi_A(R)$		
$R \bowtie S$		
$R \bowtie S$		
$R \bowtie S$		
$\Pi_B(R) \cup \Pi_B(S)$		
$\Pi_B(R) \cap \Pi_B(S)$		
$\Pi_B(R) - \Pi_B(S)$		
$R \div \Pi_B(S)$		
$\gamma_{A; count(B) \rightarrow D}(R)$		

习题 III

- ④ 设属性 K 是关系 R 的主键，写一个关系代数表达式来验证 R 的实例是否违反实体完整性约束，说明如何用该关系代数表达式的结果来完成验证。
- ⑤ 设属性 K 是关系 R 的主键，关系 S 的外键 F 参照 $R.K$ ，写一个关系代数表达式来验证 R 和 S 的实例是否违反参照完整性约束，说明如何用该关系代数表达式的结果来完成验证。
- ⑥ 在课上用的College数据库上，用关系代数查询3006号课程的最高分
 - ▶ 方法1: 只用基本关系代数操作
 - ▶ 方法2: 用外连接