

Relational Databases

哈尔滨工业大学
计算机科学与技术学院
海量数据计算研究中心
电子邮件: znzou@hit.edu.cn

2019年春

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

¹课件更新于2019年3月27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)
- 关系数据模型是众多关系数据库管理系统的模型基础

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

- 关系数据模型使用唯一的数据结构——**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

Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math

Navigation icons: back, forward, search, etc.

关系(Relation)的定义

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)**

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

221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math

$$Student \subseteq D_1 \times D_2 \times D_3 \times D_4 \times D_5$$

Navigation icons: back, forward, search, etc.

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

Student				
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232101	Abby	F	19	Math

上面的Student关系是不正确的，因为一个人不能同时有2个年龄

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 $R(A_1, A_2, \dots, A_n)$ 。

Student				
Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	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)。

- 例：在关系Grade(StudentNo, CourseNo, Score)中，属性组{StudentNo, CourseNo}和{StudentNo, CourseNo, Score}都是超键

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。

- 例：{StudentNo, CourseNo}为关系Grade的候选键

Definition (主键)

一个关系有至少一个候选键，指定其中一个作为主键(primary key)。

关系的键(Keys)

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，但 F 不是 R 的主键。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?)

Student

Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232102	Cincy	F	18	

Department

Dept	Sloc
Physics	B1
CS	B2
Math	B3

{Sdept}是Student的外键，它参照Department的主键{Dept}

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
 - ▶ 插入数据
 - ▶ 修改数据
 - ▶ 删除数据

查询语言(Query Languages)

查询语言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章第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)
 - ▶ 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

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

Student				
Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math

参照完整性约束

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

Student

Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232102	Cincy	F	18	

Department

Dept	Sloc
Physics	B1
CS	B2
Math	B3

用户定义完整性约束

根据应用需求定义的完整性约束条件

- 考试成绩在0-100分之间
- 性别必须为'M'或'F'

关系的模式(Schema)与实例(Instance)

- 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
- 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

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运算(operation)表达式来表示查询的语言
- 关系代数查询表达式明确给出了查询的执行过程
- 关系运算的三要素
 - ① 运算对象: 关系
 - ② 运算符: 选择 σ 、投影 Π 、笛卡尔积 \times 、并 \cup 、差 $-$ 、重命名 ρ 、交 \cap 、连接 \bowtie 等
 - ③ 运算结果: 关系

基本关系代数操作

六项基本关系代数操作

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

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

- ① 找出计算机系的全体学生 $\sigma_{Sdept='CS'}(Student)$
- ② 找出计算机系的全体男同学 $\sigma_{Sdept='CS' \wedge Ssex='M'}(Student)$

Student

Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math

查询1的结果

Sno	Sname	Ssex	Sage	Sdept
231101	Elsa	F	19	CS
231102	Eric	M	19	CS

查询2的结果

Sno	Sname	Ssex	Sage	Sdept
231102	Eric	M	19	CS

选择操作

关系代数在线练习 <https://dbis-uibk.github.io/relax/calc.htm>
电子产品数据库

- Product(maker, model, type)
- PC(model, speed, ram, hd, price)
- Laptop(model, speed, ram, hd, screen, price)
- Printer(model, color, type, price)

Example (练习)

- ① What PC models have a speed of at least 3.00?
- ② What PC models have a speed of at least 3.00 and ram of at least 1024MB?
- ③ What PC models have a speed of at least 3.00 or ram of at least 1024MB?

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

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

② 找出全部的系 $\Pi_{Sdept}(Student)$

Student

Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math

查询1的结果

Sno	Sname
221101	Nick
231101	Elsa
231102	Eric
232101	Abby

查询2的结果

Sdept
Physics
CS
Math

投影操作

电子产品数据库

- Product(maker, model, type)
- PC(model, speed, ram, hd, price)
- Laptop(model, speed, ram, hd, screen, price)
- Printer(model, color, type, price)

Example (练习)

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

笛卡尔积操作

- 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

Student				
Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232102	Cincy	F	18	Math

SC		
Sno	Cno	Grade
221101	1002	92
221101	1003	85
221101	1006	88
231101	1006	90
231102	1003	80
232101	1002	100

$Student \times SC$								
Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade	
221101	Nick	M	20	Physics	221101	1002	92	
221101	Nick	M	20	Physics	221101	1003	85	
221101	Nick	M	20	Physics	221101	1006	88	
221101	Nick	M	20	Physics	231101	1006	90	
221101	Nick	M	20	Physics	231102	1003	80	
221101	Nick	M	20	Physics	232101	1002	100	
...

笛卡尔积操作

Tips

- 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
- 单独计算 $R \times S$ 一般是没有什么实际意义的(为什么?)
- 笛卡尔积操作通常和选择操作一起使用, 即连接(join)

电子产品数据库

- Product(maker, model, type)
- PC(model, speed, ram, hd, price)
- Laptop(model, speed, ram, hd, screen, price)
- Printer(model, color, type, price)

Example (练习)

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

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 (练习)

电子产品数据库

- Product(maker, model, type)
 - PC(model, speed, ram, hd, price)
 - Laptop(model, speed, ram, hd, screen, price)
 - Printer(model, color, type, price)
- ① Find the model numbers and price of all PC's and all laptops

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

Example (练习)

电子产品数据库

- Product(maker, model, type)
 - PC(model, speed, ram, hd, price)
 - Laptop(model, speed, ram, hd, screen, price)
 - Printer(model, color, type, price)
- ① Find the manufacturers that sell laptops but not PC's

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

Tips

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 (练习)

- 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

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 内连接 \bowtie_{θ}
- ③ 自然连接 \bowtie
- ④ 外连接: 左外连接 \bowtie_{L} 、右外连接 \bowtie_{R} 、全外连接 \bowtie_{F}
- ⑤ 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

Property

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

Example (练习)

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

内连接(Inner Join)/ θ 连接(θ -Join)

- 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

Student				
Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232102	Cincy	F	18	Math

SC		
Sno	Cno	Grade
221101	1002	92
221101	1003	85
221101	1006	88
231101	1006	90
231102	1003	80
232101	1002	100

$Student \bowtie_{Student.Sno=SC.Sno} SC$							
Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
221101	Nick	M	20	Physics	221101	1002	92
221101	Nick	M	20	Physics	221101	1003	85
221101	Nick	M	20	Physics	221101	1006	88
231101	Elsa	F	19	CS	231101	1006	90
231102	Eric	M	19	CS	231102	1003	80
232101	Abby	F	18	Math	232101	1002	100

内连接(Inner Join)/ θ 连接(θ -Join)

Property

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

Example (练习)

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

等值连接(equi-join): 连接条件仅涉及相等比较的连接称作等值连接

自然连接(Natural Join)

- 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: $R \bowtie S$

Student				
Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232102	Cincy	F	18	Math

SC		
Sno	Cno	Grade
221101	1002	92
221101	1003	85
221101	1006	88
231101	1006	90
231102	1003	80
232101	1002	100

Student \bowtie SC						
Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
221101	Nick	M	20	Physics	1002	92
221101	Nick	M	20	Physics	1003	85
221101	Nick	M	20	Physics	1006	88
231101	Elsa	F	19	CS	1006	90
231102	Eric	M	19	CS	1003	80
232101	Abby	F	18	Math	1002	100

自然连接(Natural Join)

- 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

Property

$$R \bowtie S = \Pi_{attr(R) \cup attr(S)} (R \bowtie_{\bigwedge_{A \in attr(R) \cap attr(S)} R.A=S.A} S)$$

Example (练习)

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

左外连接(Left Outer Join)

- 目的: $R \bowtie_{\theta} S$ (内连接) 的结果只包含 R 和 S 中满足连接条件 θ 的元组, 有些情况下我们需要在连接结果中保留 R 或 (和) S 中的全部元组, 例如学校想了解学生的选课情况, 既要知道哪些学生选了哪些课, 也要知道哪些学生没选课

Student				
Sno	Sname	Ssex	Sage	Sdept
221101	Nick	M	20	Physics
231101	Elsa	F	19	CS
231102	Eric	M	19	CS
232101	Abby	F	18	Math
232102	Cincy	F	18	Math

SC		
Sno	Cno	Grade
221101	1002	92
221101	1003	85
221101	1006	88
231101	1006	90
231102	1003	80
232101	1002	100

想要的查询结果

Student.Sno	Sname	Ssex	Sage	Sdept	SC.Sno	Cno	Grade
221101	Nick	M	20	Physics	221101	1002	92
221101	Nick	M	20	Physics	221101	1003	85
221101	Nick	M	20	Physics	221101	1006	88
231101	Elsa	F	19	CS	231101	1006	90
231102	Eric	M	19	CS	231102	1003	80
232101	Abby	F	18	Math	232101	1002	100
232102	Cincy	F	18	Math			

左外连接(Left Outer Join)

- 指定 R 为左关系(left relation), S 为右关系(right relation)
- 功能:
 - 将 R 和 S 中满足给定连接条件 θ 的元组进行连接, 即计算 $R \bowtie_{\theta} S$
 - 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

Example (练习)

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

Question

为什么没有左内连接?

Navigation icons: back, forward, search, etc.

右外连接(Right Outer Join)

- 指定 R 为左关系(left relation), S 为右关系(right relation)
- 功能:
 - 将 R 和 S 中满足给定连接条件 θ 的元组进行连接, 即计算 $R \bowtie_{\theta} S$
 - 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: $R \Join_{\theta} S$
 - \Join —右外连接操作符

Example (练习)

- Execute $Product \Join PC$

Navigation icons: back, forward, search, etc.

全外连接(Full Outer Join)

- 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, 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

Property

$$R \bowtie_{\theta} S = R \bowtie_{\theta} S \cup R \bowtie_{\theta} S$$

Example (练习)

- Execute $Product \bowtie PC$

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 $yz \leq x$ 的最大的正整数 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
221101	1002
221101	1003
221101	1006
231101	1002
231101	1006
231102	1003
232101	1002

\div

Cno
1002
1006

$=$

Sno
221101
231101

除(Division)

Example (练习)

- ① What manufacturers make **all** types of products (PC, laptop, and printer)? (还有其他做法)

Question

如何用基本关系代数操作来等价地表示除操作?

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
221101	1002	92
221101	1003	85
221101	1006	88
231101	1002	95
231101	1006	90
231102	1003	80
232101	1002	100

每名学生的选课数和平均分		
Sno	Amount	AvgGrade
221101	3	83.3
231101	2	92.5
231102	1	80
232101	1	100

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 (练习)

- 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

Example (练习)

- 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?)

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 \bowtie_{\rightarrow} 、全外连接 $\bowtie_{\leftrightarrow}$)、除 \div
 - ▶ **扩展关系代数操作**: 分组操作 γ 、赋值操作 $=$
- ③ **在线练习**: <https://dbis-uibk.github.io/relax/calc.htm>

课堂练习

设关系 $R(A, B)$ 中包含 r 个元组，关系 $S(B, C)$ 中包含 s 个元组，其中 $r, s > 0$ 。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
$\sigma_{A < B}(R)$	0	r
$\Pi_A(R)$	1	r
$R \bowtie S$	0	rs
$R \Join S$	r	rs
$R \Join S$	$r + s$	rs
$\Pi_B(R) \cup \Pi_B(S)$	1	$r + s$
$\Pi_B(R) \cap \Pi_B(S)$	0	$\min(r, s)$
$\Pi_B(R) - \Pi_B(S)$	0	r
$R \div \Pi_B(S)$	0	$\lfloor r/s \rfloor$
$\gamma_{A; \text{count}(B) \rightarrow D}(R)$	1	r

问题汇编

- ① 一个关系的外键可以参照该关系自身的主键吗？

答: 可以。考虑关系 $\text{Student}(\underline{\text{Sno}}, \text{Sname}, \text{Ssex}, \text{Sage}, \text{Sdept}, \text{Mno})$ ，其中 Sno 是主键， Mno 是学生的班长的学号。因为班长也是学生，所以 Mno 是 Student 的外键，它参照 Student 的主键 Sno 。但是，一个关系的外键不能参照该外键自身。

致谢

- 感谢李治霖同学指出课件中的错误
- 感谢龚利锋、王梓宣、肖潇、李一鸣同学提供课堂练习题的笔记