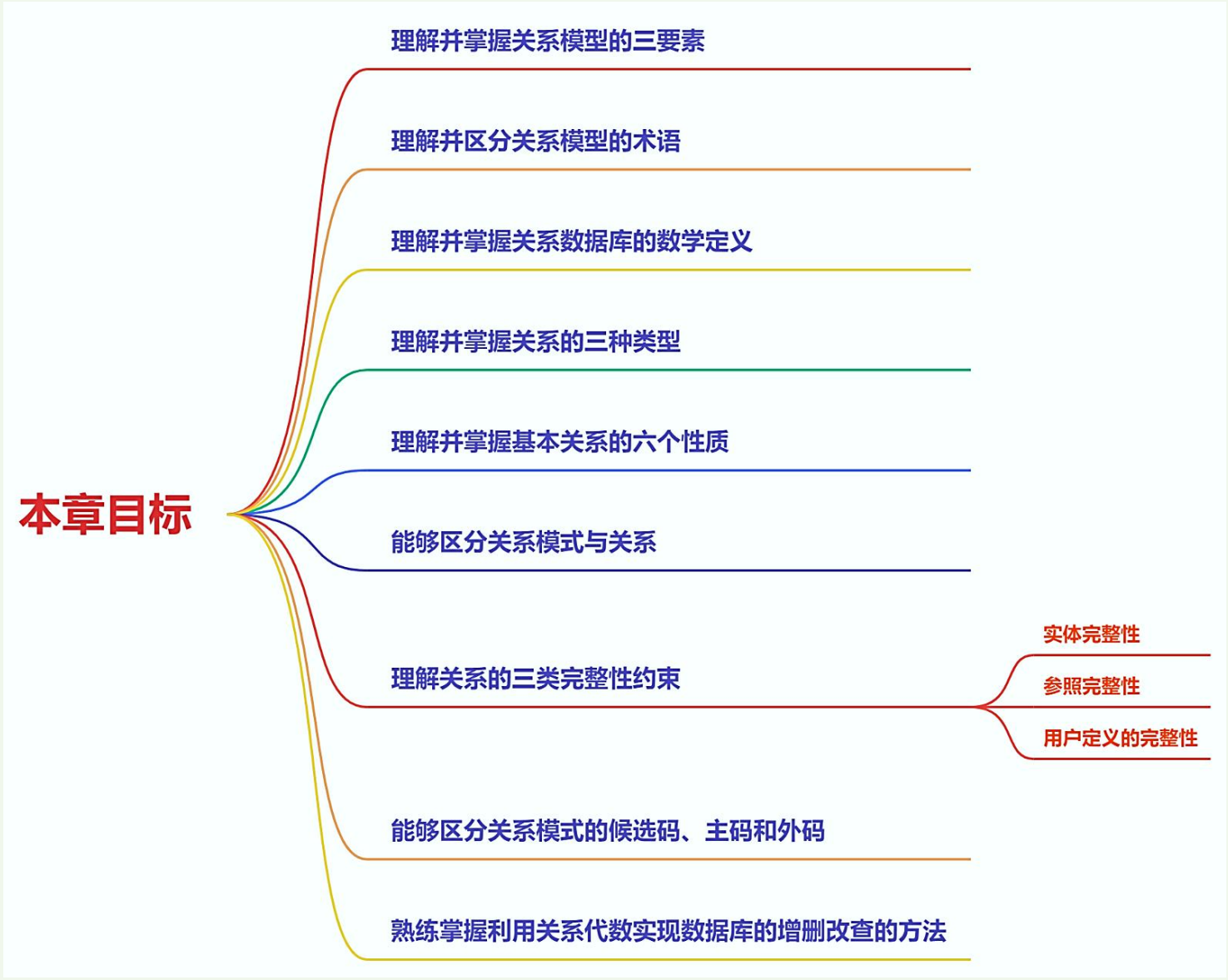
第2章　关系模型

1 本章导图


本章目标
理解并掌握关系模型的三要素
理解并区分关系模型的术语
理解并掌握关系数据库的数学定义
理解并掌握关系的三种类型
理解并掌握基本关系的六个性质
能够区分关系模式与关系
理解关系的三类完整性约束
实体完整性
参照完整性
用户定义的完整性
能够区分关系模式的候选码、主码和外码
熟练掌握利用关系代数实现数据库的增删改查的方法

2 大纲

- 关系模型
- 关系模型的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

3 关系模型

- 关系模型应用数学方法来处理数据库中的数据
- 系统严格提出关系模型的是美国IBM公司的E.F.Codd

- 1962年
- CODASYL发表“信息代数”方法用于数据处理

- 1968年
- David Child在IBM 7090机上实现了集合论数据结构

- 1970年
- E.F.Codd, A relational model of data for large shared data banks, Communication of the ACM
- ACM,自1958年以来1/4世纪中具有里程碑意义的25篇研究论文之一

- 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Oracle 1.0，并不断发展成熟

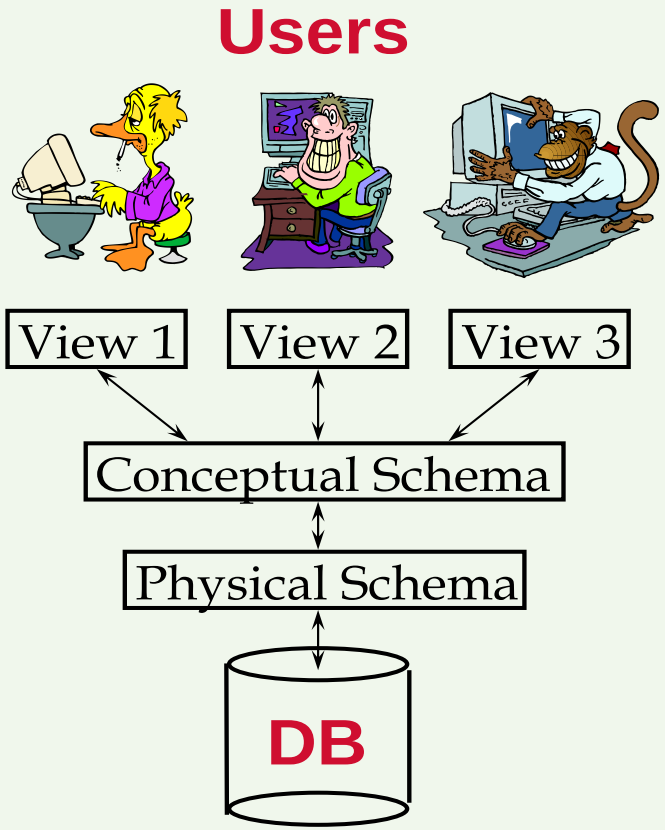
- 关系模型=关系数据结构+关系数据操作+完整性约束

4 关系模型的数据结构及形式化定义

- 关系
- 关系模式
- 关系数据库
- 关系模型的存储结构

4.1关系

- 关系模型具有单一的数据结构：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
- 关系模型中数据的逻辑结构：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
- 关系是建立在集合代数基础上的



- 域(Domain)
  - 一组具有相同数据类型的值的集合
  - 例：整数, 实数, {'男', '女'}, 字符串, 日期, ...

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

- 笛卡尔积(Cartesian Product)
  - 给定一组域$D_1, D_2, \ldots, D_n$，允许其中某些域相同，$D_1, D_2, \ldots, D_n$的笛卡尔积为：
    $D_1 \times D_2 \times \ldots \times D_n = \{(d_1, d_2, \ldots, d_n) | d_i \in D_i, i = 1, 2, \ldots, n\}$
    - 其中，每个元素$(d_1, d_2, \ldots, d_n)$称为一个n元组(n-tuple)，简称元组，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(Cardinal number)：$|D_1| \times |D_2| \times \ldots \times |D_n|$
  - 笛卡尔积可表示为一个二维表，表中行对应元组，列对应域

$D_1 = \{1, 3\}$ $D_2 = \{2, 4\}$ $D_3 = \{5, 6\}$

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

基数=2*2*2=8

- 关系(Relation)：
  - $D_1 \times D_2 \times \ldots \times D_n$的子集叫做域$D_1, D_2, \ldots, D_n$上的关系，记为$R(D_1, D_2, \ldots, D_n)$
    - 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、单目关系(Unary relation)
    - n=2：二元关系或二目关系(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

4 ~(cont.)

- 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

$$(d_1,d_2,\ldots,d_n) \neq (d_2,d_1,\ldots,d_n)$$

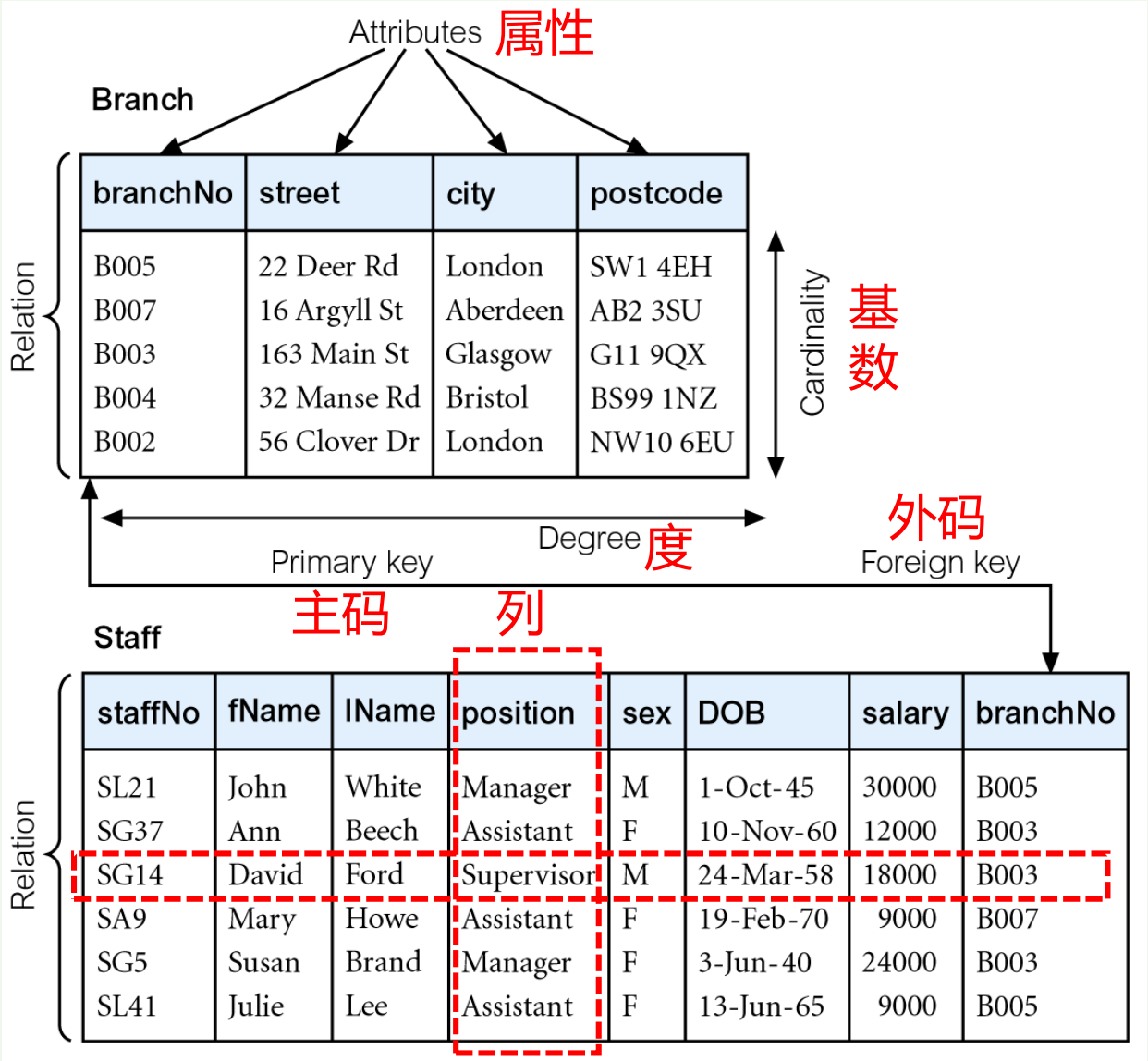
但关系满足交换律，即

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

- 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，Non-key attribute)：不包含在任何候选码的属性


Attributes 属性
Branch
branchNo | street | city | postcode
B005 | 22 Deer Rd | London | SW1 4EH
B007 | 16 Argyll St | Aberdeen | AB2 3SU
B003 | 163 Main St | Glasgow | G11 9QX
B004 | 32 Manse Rd | Bristol | BS99 1NZ
B002 | 56 Clover Dr | London | NW10 6EU
关系
Relation
Cardinality
基数
Degree 度
Primary key
主码
Foreign key
外码
Staff
列
staffNo | fName | lName | position | sex | DOB | salary | branchNo
SL21 | John | White | Manager | M | 1-Oct-45 | 30000 | B005
SG37 | Ann | Beech | Assistant | F | 10-Nov-60 | 12000 | B003
SG14 | David | Ford | Supervisor | M | 24-Mar-58 | 18000 | B003
SA9 | Mary | Howe | Assistant | F | 19-Feb-70 | 9000 | B007
SG5 | Susan | Brand | Manager | F | 3-Jun-40 | 24000 | B003
SL41 | Julie | Lee | Assistant | F | 13-Jun-65 | 9000 | B005
Relation
行

- 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
    - **注意：**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，可周期性刷新(refresh)
    - **CREATE MATERIALIZED VIEW view_table，…**

- 基本关系的六个性质：
  - 三列两行一分量（“321”）
    - 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

规范化的最基本条件

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

非规范关系

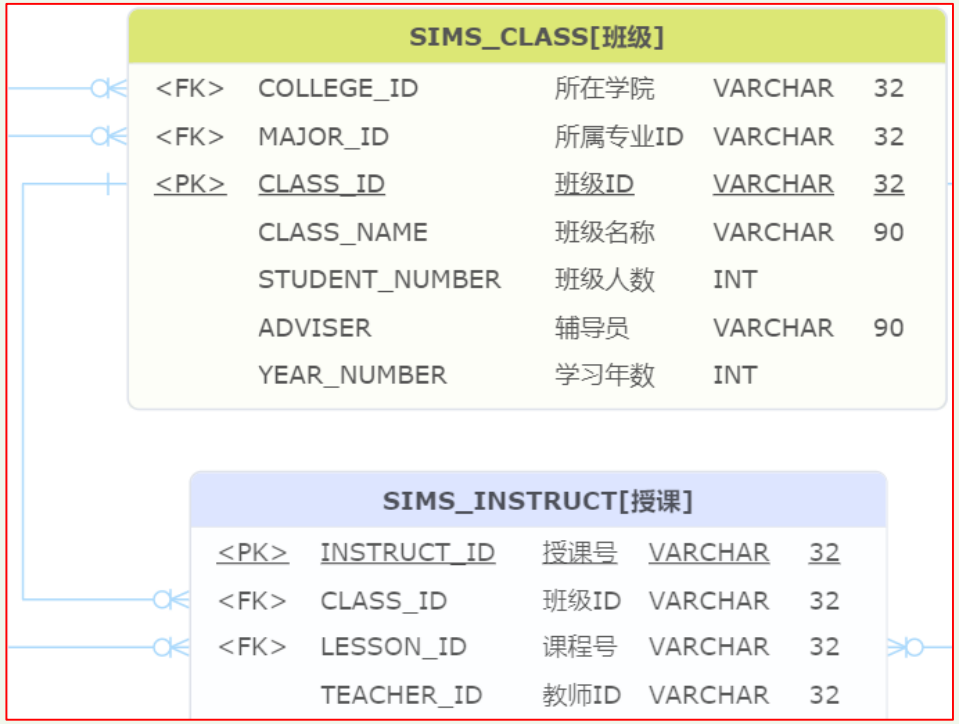
SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	计算机科学与技术	李勇	刘晨
刘逸	信息管理与信息系统	王敏	

← 小表

4.2关系模式

- 关系模式是型，关系是值
- 什么是关系模式?
  - 关系模式是对关系的描述
    - 描述了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
    - 描述了关系的完整性约束
      - 属性构成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)

示例



- 关系模式的形式化定义：
  - 五元组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：属性间的数据依赖关系集合

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

可用SQL语句 CREATE TABLE，ALTER TABLE 命令完成关系模式的创建与修改

  - 简记为R(U)，或$R(A_1,A_2,\ldots,A_n)$, A_i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

- 关系模式与关系的联系与区别

关系模式	关系
• 静态 • 稳定	• 动态 • 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：
  - 在实操过程中是先建关系模式还是先建关系？或者相反？

S001	张三	21
S002	李四	20
S003	王五	22

C001	数据库	4
C002	英语	6
C003	数学	6

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(课程号，课程名，学分)
选课(学号，课程号，成绩)

4.3关系数据库

- 支持关系模型的数据库系统称为关系数据库系统

- 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都可以分别用一个关系模式来描述

关系模式	字段含义
Student(sno, sname, ssex,sbirthdate, smajor)	学号，姓名，性别，出生日期，主修专业
Course(cno, cname, ccredit, cpno)	课程号，课程名，学分，先修课(直接先修课)
SC(sno, cno, grade, semester, teachingclass)	学号，课程号，成绩，开课学期，教学班

学生关系模式实例

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

课程关系模式实例

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

选课关系模式实例

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

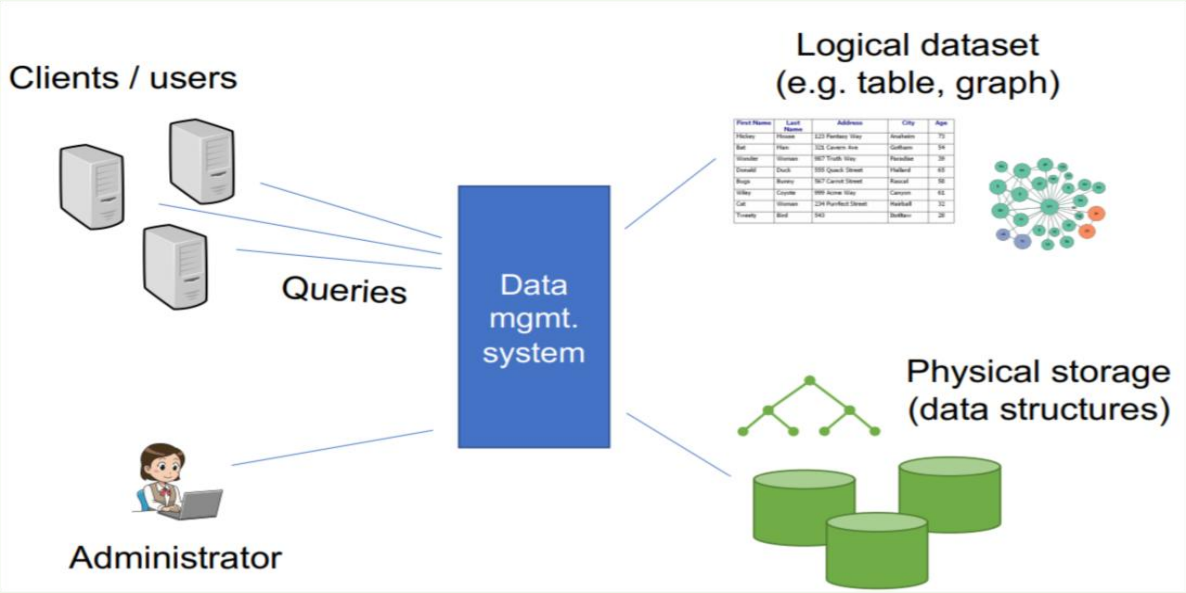
- 关系数据库也有型和值之分

- 关系数据库的型(Type)
  - 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(Instance)
  - 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

4.4关系模型的存储结构

- 关系模型是关系数据库的逻辑结构，用关系数据定义语言描述
- 关系数据库管理系统(RDBMS)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RDBMS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RDBMS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



5 关系操作

- 关系模型给出了关系操作能力的说明，但不对RDBMS语言给出具体的语法要求，即不同的RDBMS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
  - 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

- 关系数据库语言分类:
  - **关系代数、关系演算、SQL**

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 • 代表：ISBL	• 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	• 代表：SQL • 注意：因为关系代数不允许重复元组，所以当编写与关系代数等价的SQL语句时如果出现重复元组则应当在SQL语句中使用DISTINCT去重

- **关系代数、元组关系演算、域关系演算、SQL的表达能力等价**

6 关系的完整性

- 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(Constraints)
  - 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

6.1实体完整性

- 规则2.1 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(Null value)
    - 属性A可以是单一属性或属性组
      - 注：如果一张表存在异于主码的其它候选码，那么其它候选码的属性也应该非空，而不仅仅是主码的属性非空
  - 空值就是“不知道”或“不存在”或“无意义”的值

- 示例：
  - 学生(学号,姓名,性别,出生日期，主修专业), 学号为主码，不能取空值
  - 选课(学号, 课程号,成绩, 开课学期，教学班)
  - (学号，课程号)为主码，这两个属性都不能取空值
    - 学号和课程号都是主属性

- 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

6.2参照完整性

- 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2.1

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)

/*专业名和专业编号都是候选码，都可以作为主码，但此处取专业名为主码*/

- 示例2.2：学生、课程、学生与课程之间的多对多联系

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(学号，姓名，性别，出生日期，主修专业)

- 同一关系内部属性间也可能存在引用关系

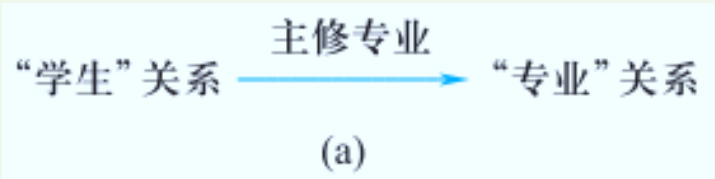
- 示例2.3：

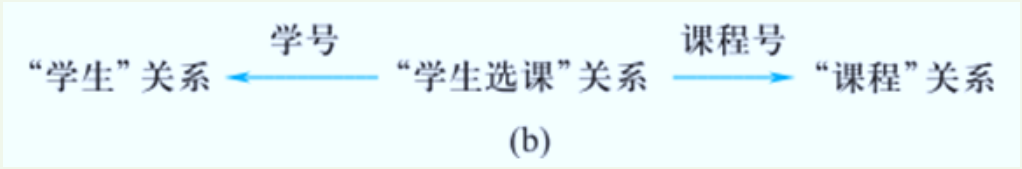
学生(学号，姓名，性别，专业号，年龄，班长)

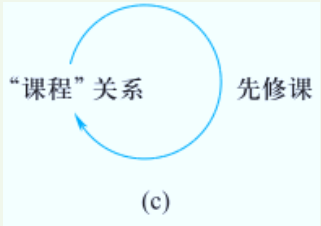
学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	

- 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
  - **说明:**
    - R和S不一定是不同的关系；S的主码K_s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







- 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

- 示例：
  - 学生关系中每个元组的“主修专业”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
  - 课程(课程号, 课程名, 学分, 先修课)中“先修课”的空值或非空取值

6.3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，以便用统一的系统方法处理它们，而不需由应用程序承担这一功能
  - 在SQL中通过CREATE/ALTER TABLE…定义
  - 定义后的约束存放在数据字典中。当需要对表进行修改操作时首先在数据字典中进行检验，通过后才能进行修改操作。

- 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

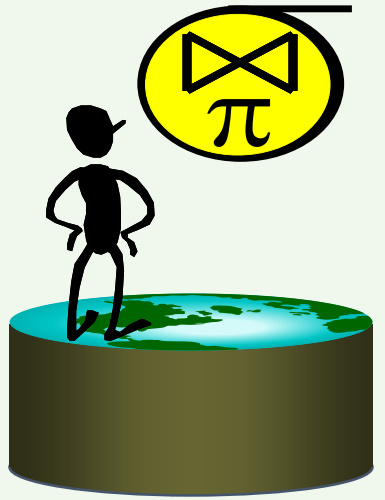
- 示例：
  - 课程(课程号，课程名，学分)
    - 课程号取唯一值；课程名非空；学分取{1, 2, 3, 4, 5}

```
--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
        Ssex     CHAR(2),
        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*/
         );
```

约束关键词:

- **PRIMARY KEY**
- **UNIQUE**
- **FOREIGN KEY**

- 关系代数是一种抽象的查询语言，它使用关系的运算来表达查询
- 运算对象、运算符、运算结果是运算的三大要素
- 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

- 关系代数运算符

运算符		含义	运算符		含义
集合运算符	∪	并	比较运算符	>	大于
	−	差		≥	大于等于
	∩	交		<	小于
	×	笛卡尔积		≤	小于等于
				=	等于
				<>	不等于

•并、差、交、广义笛卡尔积 ← 传统的集合运算

专门的关系运算 → •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	选择	逻辑运算符	¬	非
	π	投影		∧	与
	⋈	连接		∨	或
	÷	除			

7.1传统的集合运算

- 并(Union)
  - $R \cup S = \{ t \mid t \in R \vee t \in S \}$
- 差(Difference)
  - $R-S = \{ t \mid t \in R \vee t \notin S \}$
- 交(Intersection)
  - $R \cap S = \{ t \mid t \in R \wedge t \in S \} = R - (R-S)$
- 笛卡尔积
  - $R \times S = \{ \overset{\frown}{t_r t_s} \mid t_r \in R \wedge t_s \in S \}$

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

$R \cap S$

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

$R-S$

A	B	C
a_1	b_1	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

7.2专门的关系运算

- $R, t \in R, t[A_i]$
  - 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
- $A,\ t[A],\ \overline{A}$
  - 若$A = \{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\}$，其中$A_{i_1}, A_{i_2}, \ldots, A_{i_k}$是$A_1, A_2, \ldots, A_n$中的一部分，则$A$称为属性列或属性组
  - **t[A]**= (t[A_{i_1}], t[A_{i_2}], …, t[A_{i_k}])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$\{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\}$ 后剩余的属性组
- $\widehat{t_r t_s}$
  - 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，是一个$n+m$目元组

- 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中的象集为：

$$Z_x = \{ t[Z] \mid t \in R,\ t[X]=x \}$$

    - 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

- 示例：

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

- x_1在R中的象集

  $Z_{x1} = \{Z_1, Z_2, Z_3\}$,
- x_2在R中的象集

  $Zx_2 = \{Z_2, Z_3\}$,
- x_3在R中的象集

  $Zx_3 = \{Z_1, Z_3\}$

- 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

- 学生关系Student、课程关系Course和学生选课关系SC

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

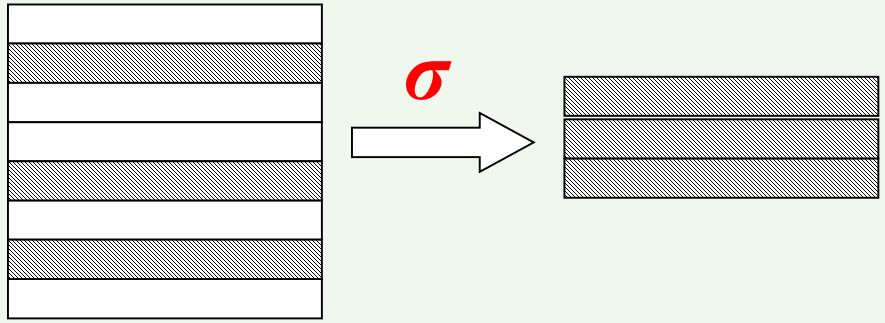
- 学生关系Student、课程关系Course和学生选课关系SC

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

7.2.1选择

- 选择(Selection)：
  - 选择又称为限制(Restriction)
  - $\sigma_F(R) = \{ t \mid t \in R \wedge F(t) = \text{'真'} \}$
  - 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：$X_1 \theta Y_1$
    - X_1，Y_1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**

[例2.4] 查询信息安全专业全体学生

$\sigma_{Smajor='信息安全'}(Student)$

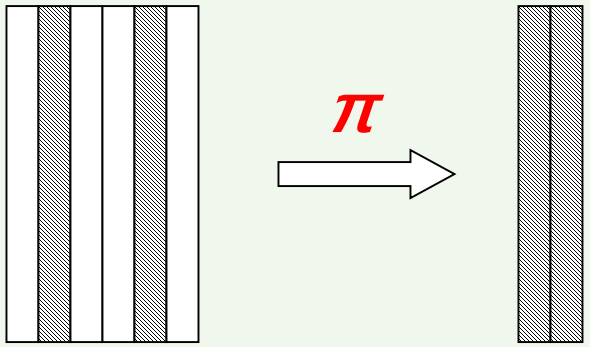
Sno	Sname	Ssex	Sbirthdate	Smajor
20180001	李勇	男	2000-3-8	信息安全

[例2.5] 查询2001年之后（包括2001年）出生的学生

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

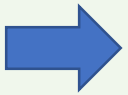
Sno	Sname	Ssex	Sbirthdate	Smajor
20180003	王敏	女	2001-8-1	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

- 投影(Projection)：
  - $\pi_A(R) = \{ t[A] \mid t \in R \}$
  - A：R 中的属性列
  - 从R 中选择出若干属性列组成新的关系
  - 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)



[例2.6] 查询学生的学号和主修专业

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



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

[例2.7] 查询学生关系Student中都主修了哪些专业

$\pi_{Smajor}(Student)$

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

7.2.3连接

- 连接(Jion)：
  - 也称为θ连接或内连接
  - $R \underset{A\theta B}{\bowtie} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B] \}$
  - 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
    - 连接运算是从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

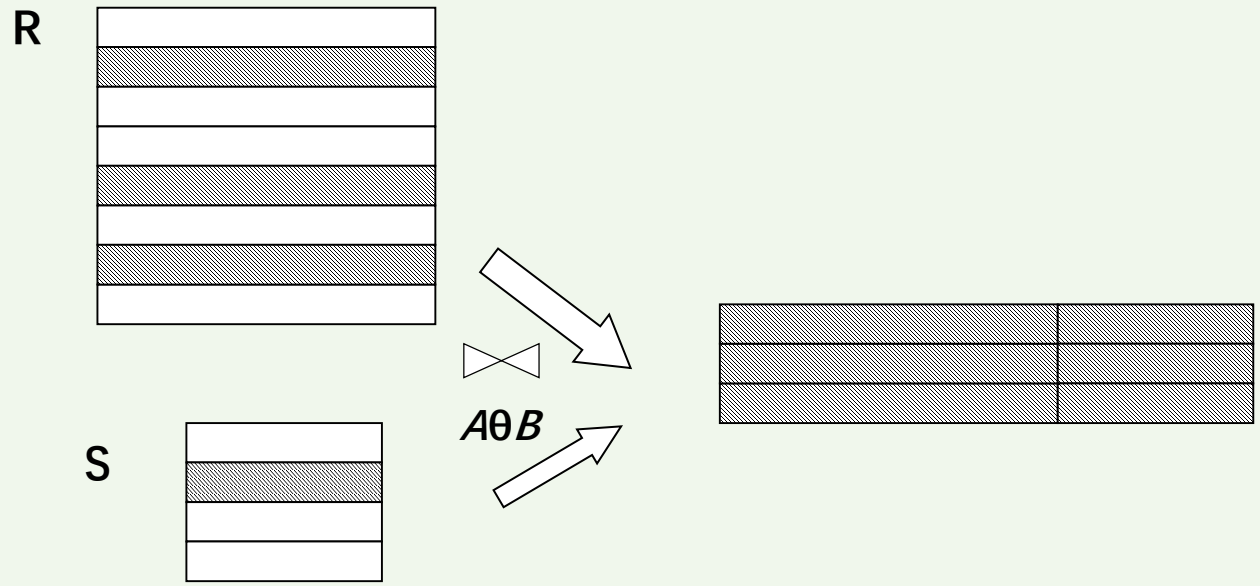
- 两类常用连接：
  - 等值连接(Equijoin)
    - θ为“=”的连接运算称为等值连接

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

  - 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

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

- 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



[例2.8]

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

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

- 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

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

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

$R \bowtie S$

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

$R \bowtie S$

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

$R \bowtie S$

- 问题：(内)连接-外连接的作用?

Employee 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;
```

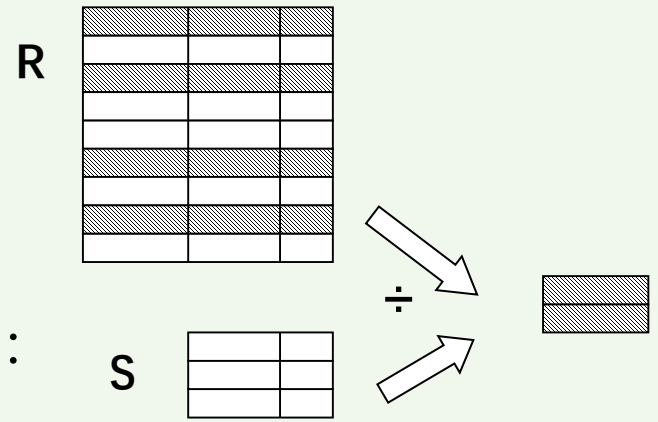
Example of a left outer join

```
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

7.2.4除

- 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：
    - $R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$
    - Y_x：x在R 中的象集，$x = t_r[X]$



除操作是同时从行和列角度进行运算

[例2.9]

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)

S在(B, C)上的投影为：

$\{(b_1,c_2), (b_2,c_1), (b_2,c_3)\}$

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$\{(b_6, c_6)\}$

- 除的实际应用场景：
  - 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”

[例2.10] 查询至少选修81001号课程和81003号课程的学生号码

【解】首先建立一个临时关系K=

Cno
81001
81003

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

[例2.11] 查询选修了81002号课程的学生的学号

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

[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

【解】 $\pi_{sname}\left(\sigma_{cpno='81003'}\left(Course \bowtie SC \bowtie \pi_{sno,\,sname}(Student)\right)\right)$

$\pi_{sname}(\sigma_{cpno='81003'}(course) \bowtie SC \bowtie \pi_{sno,sname}(student))$

$\pi_{sname}\left(\pi_{sno}\left(\sigma_{cpno='81003'}(Course)\right) \bowtie SC\right) \bowtie \pi_{sno,\,sname}(Student)\Big)$

[例2.13] 查询选修了全部课程的学生的学号和姓名

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

7 关系代数小结

- 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种基本运算来表达
    - 引进它们不增加语言的运算能力，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

8 本章小结

- 关系模型是关系数据库的核心和基础，关系数据库系统是目前使用最广泛的数据库系统

关系数据结构	关系操作	关系的完整性
• 关系 – 域 – 笛卡尔积 – 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	• 数据查询 – 选择、投影、连接、除、并、交、差、笛卡儿积 • 数据更新 – 插入 – 删除 – 修改 • 关系代数表达式 • 关系代数是一种过程性语言，区分于SQL	• 实体完整性 • 参照完整性 – 外码 • 用户定义的完整性

9 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. 3 **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(职工号，姓名，年龄，部门号)和部门(部门号，部门名称)存在参照关系，其中_____是_____参照关系，是外码。

10 本章作业

- 教材第二章全部习题，其中第6题只需完成关系代数的查询
- 要求：作业在布置后一周内完成并提交到课程网站