

Database System

Chapter 2 The Relational Model



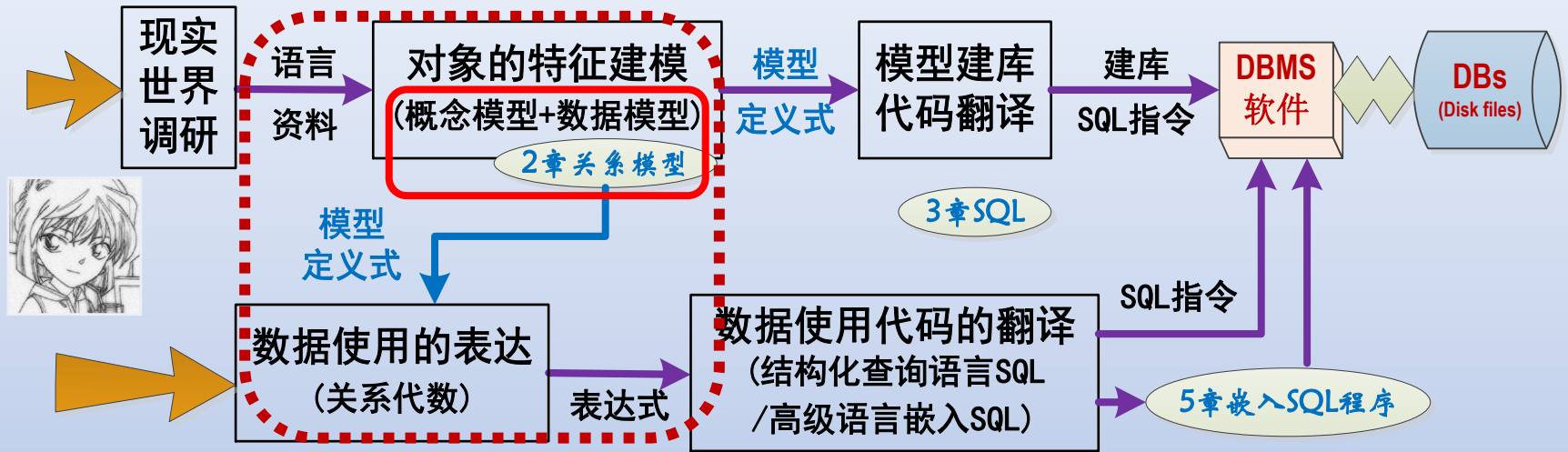
**School of Information
and Control**

孔月萍,贺秦禄,王佳婧

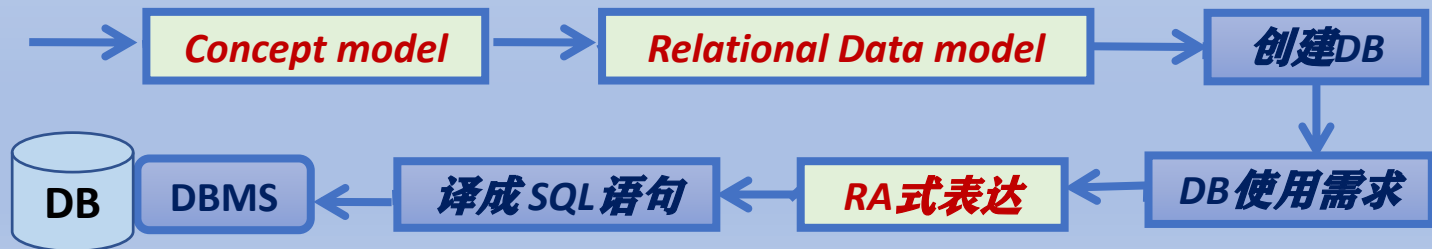


Studying Questions

Chapter 2 : The Relational Model



• How to *Build a DB in Logical* and how to *use it*?





Database Principles,

Programming and Performance

Chapter 2 the Relational Model

2.1 CAP Database

2.2 Naming the parts of a Database

2.3 Relational Rules

2.4 Keys, Superkey and Null Values

关系概念

2.5 Relational Algebra

2.6 Set-Theoretic Operations

2.7 Native Relational Operations

2.8~2.10 Other Relational operations

2.9 Illustrative Examples

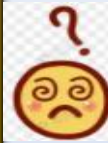
关系代数





2.1 The CAP Database

- What is the *Relational Data Model* ?
- How to *use* it? ---- *Relation Algebra*



Customers			
cid	cname	city	dicsnt
c001	Tip Top	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00

1970年, IBM研究员E.F.Codd博士发表文章:

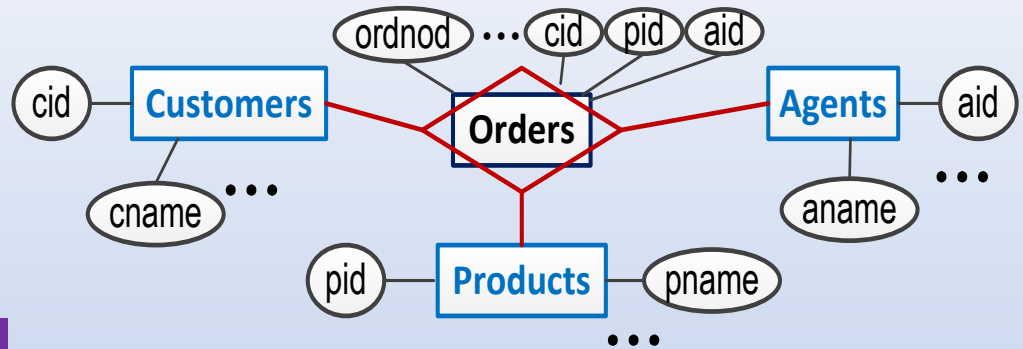
----A Relational Model of Data for Large Shared Data Banks

- 现实世界的各种实体、实体间联系均可用 **关系模型(Table)** 表示。
- 关系模型是在对Table及其操作进行数学化定义基础上提出的, 借助于 **集合代数、逻辑学** 等数学概念和方法来处理table的数据。 **关系模型** 由 “**关系数据结构、关系操作集、关系完整性约束**” 三部分组成。
- *Relational Data Model* 是数据库的四大经典模型之一, 今天的大多数商品化DBS仍使用它。标准的 **数据库语言SQL** 就建立在关系模型之上。



2.1 The CAP Database

1. Concept Model



2. Relational Data Model

Structure of *CAP Database* (p27~28) With *Four Tables*.

- **CAP** = {customers, agents, products, orders}
- Head(customers) = {cid, cname, city, disct}
- Head(agents) = {aid, aname, city, percent}
- Head(products) = {pid, pname, city, quantity, price}
- Head(orders) = {ordno, month, cid, aid, pid, qty, dollars}

Data model
Components

3. Relational Algebra (关系代数)

- **Use** Tables of Database, and obey some **Constraints**
- 简写RA. 任何RDBMS上, 对数据查询的数学表达式 (一种代数语言).



2.1 The CAP Database

CAP_(Database Schema)

②... **View / Trigger / Index / User / ...**
(Database Objects)

① **Table / Relation**

Relational Schema

Row / Record / Tuple
(Data set, elements)

Column / Attribute / Field

ColumnName, Datatype, Domain
Column value

Customers			
cid	cname	city	dicsnt
c001	Tip Top	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c005	ACME	Kyoto	0.00

Products				
pid	pname	city	quantity	price
p01	comb	Dallas	111400	0.50
p02	brush	Newark	203000	0.50
p03	razor	Duluth	150600	1.00
p04	pen	Duluth	125300	1.00
p05	pencil	Dallas	221400	1.00

Candidate key / Table key

└─ **Primary key / Super key**

└─ **Primary attribute / Non key attribute**

ODateTime		Orders				
ordno	month	cid	aid	pid	qty	dollars
1011	jan	c001	a01	p01	1000	450
1019	feb	c001	a02	p02	400	180
1017	feb	c001	a06	p03	600	540
1013	jan	c002	a03	p03	1000	880
1026	may	c002	a05	p03	800	704

Agents			
aid	cname	city	percent
a01	Smith	New York	6
a02	Jones	Newark	6
a03	Brown	Tokyo	7
a04	Gray	New York	6
a05	Otasi	Duluth	5



2.1 The CAP Database

[1] Database Database schema = {table-1, table-2, ..., table-n}

CAP = {customers, agents, products, orders}

[2] Table Head name = {C1, C2,, Cn}

Customers = {cid char(4), cname char(10), city char(30), discnt int}

.....

- Meanings of column-names (P27)
- Primary Key: *cid* is unique for customers, *cname* is NOT. (*aid*, *ordno*)
- Calculate column: *dollars* = qty * price * (100 - discnt)/100

[3] Domains & Data Types

- Data Type: *Ci* must define *Data Type* such as real, int, char etc.

the Data Types are not same in different DBMS.

- Domain: *Field-Value* must be (1) Same Data Type (2) Set of value
✓ 大多数DBMS不支持枚举类型. Eg. {0, 1, 3, 4, 7} or {b, f, r, w}



2.1 The CAP Database

HIT 第3讲

关系模型简述?

(2)关系模型研究什么



- 形象地说, 一个关系(relation)就是一个Table
- 关系模型就是处理Table的, 它由三个部分组成:
 - 描述DB各种数据的基本结构形式(Table/Relation)
 - 描述Table与Table之间所可能发生的各种操作(关系运算)
 - 描述这些操作所应遵循的约束条件(完整性约束)
- 就是要学习: Table如何描述, 有哪些操作、结果是什么、有哪些约束等?

学生登记表

学号	姓名	性别	出生年月	入学日期	家庭住址
98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
98110102	张四	女	1980.04	1998.09	吉林省长春市
98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市
98110201	王三	男	1980.06	1998.09	辽宁省沈阳市
98110202	王四	男	1979.01	1998.09	山东省青岛市
98110203	王武	女	1981.06	1998.09	河南省郑州市

操作

有哪些?

学生成绩单

班级	课程	教师	学期	学号	姓名	成绩
981101	数据库	李四	98秋	98110101	张三	100
981101	数据库	李四	98秋	98110102	张四	90
981101	数据库	李四	98秋	98110103	张五	80
981101	计算机	李五	98秋	98110101	张三	89
981101	计算机	李五	98秋	98110102	张四	98
981101	计算机	李五	98秋	98110103	张五	72
981102	数据库	李四	99秋	98110201	王三	30
981102	数据库	李四	99秋	98110202	王四	90
981102	数据库	李四	99秋	98110203	王武	78

结果

是什么?



2.1 The CAP D

HIT 第3讲

为什么把“表”称为关系？
怎样严格定义一个“表”？
“表”和“关系”有什么异同？

什么是关系？

(2)“表”的严格定义--关系？

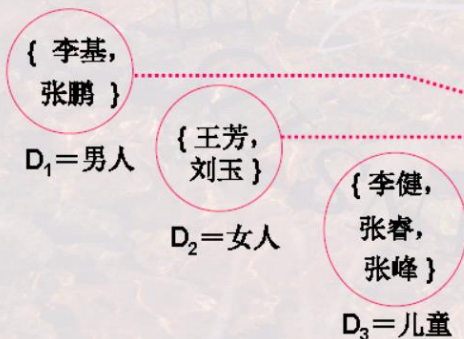
➤再定义“元组”及所有可能组合成的元组：笛卡尔积

笛卡尔积(Cartesian Product)

□一组域 D_1, D_2, \dots, D_n 的笛卡尔积为：

$$D_1 \times D_2 \times \dots \times D_n = \{ (d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, \dots, n \}$$

□笛卡尔积的每个元素 (d_1, d_2, \dots, d_n) 称作一个n-元组 (n-tuple)



笛卡尔积

男人	女人	儿童
李基	王方	李健
李基	王方	张睿
李基	王方	张峰
李基	刘玉	李健
李基	刘玉	张睿
李基	刘玉	张峰
张鹏	王方	李健
张鹏	王方	张睿
张鹏	王方	张峰
张鹏	刘玉	李健
张鹏	刘玉	张睿
张鹏	刘玉	张峰

什么是关系？

(3)“表”的严格定义--关系？

➤首先定义“列”的取值范围“域(Domain)”

域(Domain)

- 一组值的集合，这组值具有相同的数据类型
- 如整数的集合、字符串的集合、全体学生的集合
- 再如，由8位数字组成的数字串的集合，由0到100组成的整数集合
- 集合中元素的个数称为域的**基数(Cardinality)**

家庭

丈夫	妻子	子女
李基	王方	李健
张鹏	刘玉	张睿
张鹏	刘玉	张峰

$D_3 = \text{儿童集合(CHILD)} = \{\text{李健, 张睿, 张峰}\}$
 $D_2 = \text{女人集合(WOMAN)} = \{\text{王芳, 刘玉}\}$
 $D_1 = \text{男人集合(MAN)} = \{\text{李基, 张鹏}\}$

One Column table

---关系模型(1列)

Cartesian Product

---关系表的组合运算



2.1 The CAP Da

HIT 第3讲

为什么把“表”称为关系？
怎样严格定义一个“表”？
“表”和“关系”有什么异同？

什么是关系？

(2)“表”的严格定义--关系？

- 元组(d_1, d_2, \dots, d_n)的每一个值 d_i 叫做一个**分量**(component)
- 元组(d_1, d_2, \dots, d_n)是从每一个域任取一个值所形成的一种组合，笛卡尔积是所有这种可能组合的集合，即：笛卡尔积是由 n 个域形成的**所有可能的** n -元组的集合

- 若 D_i 的基数为 m_i ，则笛卡尔积的**基数**，即元组个数为

$$m_1 \times m_2 \times \dots \times m_n$$

笛卡尔积

男人	女人	儿童
李基	王方	李键
李基	王方	张睿
李基	王方	张峰
李基	刘玉	李键
李基	刘玉	张睿
李基	刘玉	张峰
张鹏	王方	李键
张鹏	王方	张睿
张鹏	王方	张峰
张鹏	刘玉	李键
张鹏	刘玉	张睿
张鹏	刘玉	张峰

域名

域值

什么是关系？

(2)“表”的严格定义--关系？

由于笛卡尔积中的所有元组并不都是有意义的，因此...

关系(Relation)

- 一组域 D_1, D_2, \dots, D_n 的笛卡尔积的子集
- 笛卡尔积中具有某一方面意义的那些元组被称作一个**关系(Relation)**
- 由于关系的不同列可能来自同一个域，为区分，需要为每一列起一个名字，该名字即为**属性名**。

笛卡尔积

男人	女人	儿童
李基	王方	李键
李基	王方	张睿
李基	王方	张峰
李基	刘玉	李键
李基	刘玉	张睿
李基	刘玉	张峰
张鹏	王方	李键
张鹏	王方	张睿
张鹏	王方	张峰
张鹏	刘玉	李键
张鹏	刘玉	张睿
张鹏	刘玉	张峰

列名(属性名)

丈夫	妻子	子女
李基	王方	李键
张鹏	刘玉	张睿
张鹏	刘玉	张峰

列值：来自域

Cartesian Product
--- 关系表的组合运算

Cartesian Product
--- 关系表的语义组合
(笛卡尔积的Subset)



2.1 The CA

HIT 第3讲

为什么把“表”称为关系？
怎样严格定义一个“表”？
“表”和“关系”有什么异同？

什么是关系？

(2)“表”的严格定义--关系？

关系模式与关系

- 同一关系模式下，可有很多的关系
- 关系模式是关系的结构，关系是关系模式在某一时刻的数据
- 关系模式是稳定的；而关系是某一时刻的值，是随时间可能变化的

Student(S# char(8), Sname char(10), Ssex char(2), Sage integer, D# char(2), Sclass char(6))

Time1: Table/Relation

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

什么是关系？

(2)“表”的严格定义--关系？

□ 关系模式 $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ 中属性向域的映象在很多DBMS中一般直接说明为属性的类型、长度等

□ 例如：

Student(S# char(8), Sname char(10), Ssex char(2), Sage integer, D# char(2), Sclass char(6))

再如：

Course (C# char(3), Cname char(12), Chours integer, Credit float(1), T# char(3))

SC(S# char(8), C# char(3), Grade float(1))

Time2: Table/Relation

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402

Relational Schema

漏排，需补齐

9月30日，5周一第3-4节，16-309，计算机0102班

9月19日，3周四第3-4节，13-316，计算机0304班

9月30日，5周一第7-8节，13-312，计算机0304班

10月10日，6周四第1-2节，13-218，计算机0304班



2.2 Naming the Parts of a Database

[4] Tables and Relations

Cartesian Product: 设 S_1, \dots, S_n 为集合, 它们的笛卡尔积为

$$S_1 \times S_2 \times \dots \times S_n = \{ (e_1, e_2, \dots, e_n) \mid e_i \in S_i, i=1, \dots, n \}$$

- Element (e_1, e_2, \dots, e_n) called a ***n-tuple*** (**n元组**)
- Value “***ei***” in the element called a **分量**

- **Relation**: $S_1 \times S_2 \times \dots \times S_n$ 的子集, 称为关系

记成 $R = S_1 \times S_2 \times \dots \times S_n$

- **Example1**: $CP = CID \times CNAME \times CITY \times DISCNT$

其中有 $T_1 = (c003, Allied, Dallas, 8.00)$ $T_2 = (c001, Basice, Oshkosh, 13.20)$

- **Example2**: **Family** = women \times men \times child

not all the tuples consist of a **real exist family**

n Column Set

Cartesian Product

Table

for Relational Meaning

Relation (schema)

关系名	列定义	约束	行
-----	-----	----	---

关系(模式)-静态 关系实例-动态



2.2 Naming the Parts of a Database

Note: $R = S_1 \times S_2 \times \dots \times S_n$

- R---关系名
- n---关系的目或度(degree). $n=1$ 一元关系, $n=2$ 二元关系
- 每列单独命名---Attribute/Field/Column, n 元关系有 n 个列
- 若某属性组的值能唯一标识关系R中的一个Tuple/Record/Row, 则该属性组为Candidate Key。仅当R中所有列构成的属性组才是它的候选码时, 称该码为All-Key(全码)。
- 若R有多个Candidate Key, 选包含属性较少的做Primary Key(主码), 候选码中的诸属性为Prime Attribute(主属性)。Non Key Attribute(非码属性)。
- Example: `orders={ ordno, month, cid, aid, pid, qty, dollars }`



2.3 Relational Rules

● Which *Property* the Relational Model should be?



[1] Rule 1. First Normal Form rule.

◆ In Relational Model, it *can not have*

(1) multi-valued fields and (2) structured fields

◆ *Example*: P33, Fig 2.3~2.5

✓ Solution-1: update the table with *new columns to fit in duplicates on different columns*

✓ Solution-2: *to create two tables and join them* in later queries

✓ ORDBMS特例:

In OR model, Figure 2.3 is OK.

So the new Object-Relational DB products break rule 1



Case1: Multi-value

Solution 2 :

Primary key

EMPLOYEES (original)

eid	ename	position	dependent
e001	Smith, John	Agent	Michael J.
			Susan R.
e002	Andrews, David	Superintendent	David M. Jr.
e003	Jones, Franklin	Agent	Andrew K.
			Mark W.
			Louisa M.

Foreign key

DEPENDENTS

eid	dependent
e001	Michael J.
e001	Susan R.
e002	David M. Jr.
e003	Andrew K.
e003	Mark W.
e003	Louisa L.

C
u
t

Solution 1 :

EMPLOYEES			dependent1	dependent2	...
eid	ename	position			
e001	Smith, John	Agent	Michael J.	Susan R.	...
e002	Andrews, David	Superintendent	David M. Jr.
e003	Jones, Franklin	Agent	Andrew K.	Mark W.	...

Case2: structured field

sname	
lname	fname
Simith	Chen
mary	Su



sname
Simith Chen
Mary Su

关系模型的“行/列”
位置不敏感



2.3 Relational Rules

[2] Rule 2. Access rows by content only.

- Can't say “the third row from the top”. (No order to the *rows* or the *columns*)
- 行标识: Disallows "pointers or Row IDs " to rows.
(most RDBMS break this rule by getting rows by RIDs)

[3] Rule 3. Unique rows.

- 2 rows can not be same in all attributes.

[4] Rule 4. Entity Integrity Rule.

- No column belonging to a primary key of table T is allowed to take on null values for any row in T.

[5] Def 2.4.1. **Table Key** 又称 **Candidate key** , 唯一标识所在行的属性组

[6] Super-key

The value of a fields group to identify a unique tuple.

If not necessarily, the **smallest** fields group become **Super-key**. ↔ **Primary key**



2.3 Relational

HIT 第3讲

(3)关系的特性?

- 理论上, **关系**的任意两个元组不能完全相同。(集合的要求: 集合内不能有相同的两个元素); **现实应用中, 表(Table)**可能并不完全遵守此特性。
- 元组相同是指两个元组的每个分量都相同。

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402
98030103	张五	男	19	03	980301

(4)关系上的一些重要概念—候选码/候选键

- 有时, 关系中有很多组候选码, 例如:

学生(S#, Sname, Sage, Sclass, Saddress)

其中属性S#是候选码, 属性组(Sname, Saddress)也是候选码(同名同地址的两个同学是不存在的)

- 再如

Employee(EmpID, EmpName, Mobile)

每一雇员有唯一的EmpID, 没有两个雇员有相同的手机号Mobile, 则EmpID是候选码, Mobile也是候选码

在同一个关系中存在相同的元组, 去掉其中的重复元组



2.3 Relational R

HIT 第3讲

(6)关系上的一些重要概念—主属性与非主属性

主属性与非主属性

- 包含在**任何一个**候选码中的属性被称作主属性，而其他属性被称作非主属性
 - 如“选课”中的S#, C#为主属性，而Sname, Cname, Grade则为非主属性；
- 最简单的，候选码只包含一个属性
- 最极端的，所有属性构成这个关系的候选码，称为全码(All-Key)。
 - 比如：关系“教师授课”(T#,C#)中的候选码(T#,C#)就是全码。

(5)关系上的一些重要概念—主码/主键

主码(Primary Key)/主键

- 当有多个候选码时，可以选定一个作为主码。
- DBMS以主码为主要线索管理关系中的各个元组。
- 例如可选定属性S#作为“学生”表的主码，也可以选定属性组(Sname, Saddress)作为“学生”表的主码。选定EmpID为Employee的主码。

(7)关系上的一些重要概念—外码/外键

外码(Foreign Key)/外键

- 关系R中的一个属性组，它不是R的候选码，但它与另一个关系S的候选码相对应，则称这个属性组为R的**外码**或**外键**。
- 例如“合同”关系中的客户号不是候选码，但却是外码。因它与“客户”关系中的候选码“客户号”相对应。
- 两个关系通常是靠**外码**连接起来的。

合同号	合同名称	合同签订人	客户号
HT0001	购煤合同	张三	CUST01
HT0002	销售机床合同	李四	CUST01
HT0003	购钢材合同	张五	CUST02

客户号	客户名称	客户地址	联系人
CUST01	依兰煤矿	哈尔滨市	王三
CUST02	长春电机厂	长春市	赵六
CUST03	鞍钢集团	鞍山市	钱七

主码：合同号 (合同表), 客户号 (客户表)

外码：客户号 (合同表)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

D#	Dname	Dean
01	机电	李三
02	能源	李四
03	计算机	李五
04	自动控制	李六

主码：S# (Student表), D# (Dept.表)

外码：D# (Student表)



2.4 Keys, Super key, Null Values

[1] Example

- CAP database, keys are columns: P27
cid, aid, pid, ordno.
- Example 2.4.1 → **AB, AC, or BC** P38

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d1
a2	b2	c1	d1
a2	b1	c2	d1

[2] Theorem 2.4.2. Every table T has **at least one key**.

[3] Def. 2.4.3. Primary Key

A **Primary Key** of a table T is the candidate key **chosen by the DBA** to uniquely identify rows in T.

So, Table-Key == Candidate-Key →→ Super-Key == Primary Key

[4] Null Values

A null value is placed in a field of a table, when the value is either **unknown** or **not yet defined**.

Data model Components: (1) [✓]Structure (2) [?]Use (3) [?]Constraint

(关系)数据结构、(关系代数)操作、(关系)完整性约束



2.5 Relational Algebra

● How to use the Relational Tables? *Relation Algebra*



Two types of RA operations:

1970年, IBM 的 E.F.Codd博士提出 (简记RA)

- **Set Theoretic** (depend on fact that *table is a set of rows*) 4种
交, 并, 差, 笛卡尔积 ---- intersection, union, difference, cartesian product

(1)集合操作

→ UNION (并)	R	S	$R \cup S$
INTERSECTION (交)	R	S	$R \cap S$
→ DIFFERENCE (差)	R	S	$R - S$
→ Cartesian PRODUCT (笛卡尔积)	R	S	$R \times S$

- **Native operations** (depend on *structure of tables*) 4种
选择, 投影, 连接, 除 ---- projection, selection, join, division

(2)纯关系操作

PROJECT (投影) ←	R		$\pi_A(R)$
SELECT (选择) ←	R		$\sigma_{Con}(R)$
JOIN (连接)	R	S	$R \bowtie_{A \theta B} S$
DIVISION (除)	R	S	$R \div S$

基本RA: 并, 差, 积, 选择, 投影

Eg: $R \cap S = (R \text{ 选 } t) \text{ and } (S \text{ 选 } t)$

$R[A1, A2, \dots]$

$R \text{ where } C$

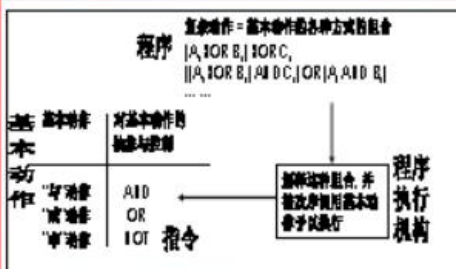


2.5 Relational Algebra

HIT 第4讲 为何要提出关系代数?

复杂动作 = 基本动作的各种方式的组合
程序 (A; XOR B;) XOR C;

基本动作
“与”动作
“或”动作
“非”动作



Select Sname From Student, SC
Where Student.S# = SC.S# and SC.C# = '001'
Order By Score DESC;

SQL语言

程序

$\Pi_{\text{Sname}}(\sigma_{\text{student.s\#}=\text{sc.s\#}}(\text{Student} \times \text{SC}))$

复杂动作 = 基本动作的各种方式的组合

关系模型基本运算的各种组合

基本动作	对基本动作的抽象与控制
“并”动作	U
“差”动作	-
“积”动作	x
“选择”动作	σ
“投影”动作	Π

指令

关系模型
基本运算

解释这种组合, 并按次序调用基本动作予以执行

数据库管
理系统

程序
执行
机构

CPU有“加
高级程序
均可用它:



2.6 Set-Theoretic Operations

1. Def. 2.6.1. Compatible Table (相容表)

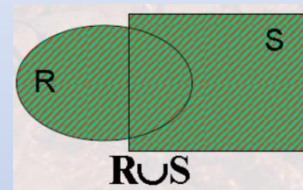
iff = If and only if

tables are said to be **compatible** iff they have the same schema.

2. Set Theoretic Operations: *R & S are compatible tables*

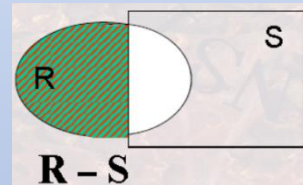
- **Intersection**: 生成n元关系, 由属于R也属于s的元组组成

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



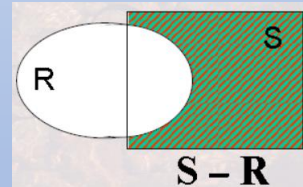
- **Union**: 生成n元关系, 由属于R或属于s的元组组成

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



- **Difference**: 生成n元关系, 由属于R而不属于s的元组组成

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



- **Extended Cartesian Product** (广义笛卡尔积):

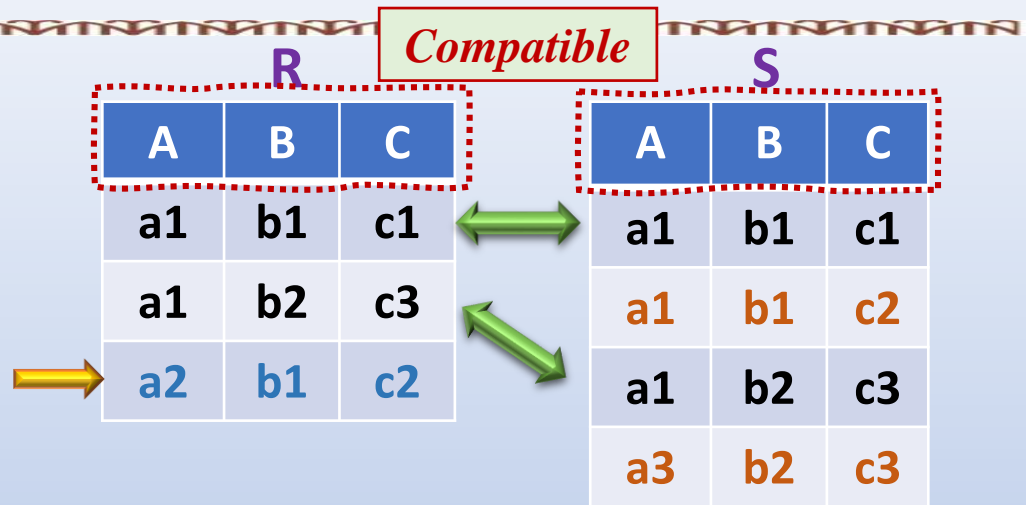
R 与 S 可不相容, 生成 m+n 列关系, 共有 $m \times n$ 个元组

$$R \times S = \{ tr \sim ts \mid tr \in R \wedge ts \in S \}$$



2.6 Set-Theoretic Operations

◆ Example 2.6.1: p43



R ∪ S

A	B	C
a1	b1	c1
a1	b2	c3
a2	b1	c2
a1	b1	c2
a3	b2	c3

R ∩ S

A	B	C
a1	b1	c1
a1	b2	c3

R - S

A	B	C
a2	b1	c2

R × S

A	B	C	A	B	C
a1	b1	c1	a1	b1	c1
a1	b1	c1	a1	b1	c2
a1	b1	c1	a1	b2	c3
a1	b1	c1	a3	b2	c3
a1	b2	c3
...

Notice: $R \cap S = S \cap R$ $R \cup S = S \cup R$ $R - S \neq S - R$ $R \times S = S \times R$ (交换律)



2.6 Set-Theoretic Operations

◆ Example

关系模型: **R(参加体育队的学生)**

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

S(参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301

R ∪ S(或者参加体育队或者文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201

$R \cap S$ (同时加入了体育队和文艺队的同学)

$R - S$ (参加体育队而未参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402



2.6 Set-Theoretic Operations

3. Assignment & Alias

- Def 2.6.3 **Assignment, Alias:**

- ✓ $\text{Head}(\mathbf{R}) = A_1, \dots, A_n.$

- ✓ create table S with $\text{Head}(\mathbf{S}) = B_1, \dots, B_n$

which has the **SAME CONTENT** as \mathbf{R} .

- ✓ Can define table \mathbf{S} by **assignment** operator

$\mathbf{S}(B_1, \dots, B_n) := \mathbf{R}(A_1, \dots, A_n)$

- **Alias:** want a different table names with same attributes

$\mathbf{S} := \mathbf{R}$

- Save intermediate results. $\mathbf{T} := (\mathbf{R} \cap \mathbf{S}) - (\mathbf{R} \infty \mathbf{S})$

or $\mathbf{T}_1 := (\mathbf{R} \cap \mathbf{S}) \quad \mathbf{T}_2 := (\mathbf{R} \infty \mathbf{S}) \quad \mathbf{T} := \mathbf{T}_1 - \mathbf{T}_2$