西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY 计算机学院 SCHOOL OF COMPUTER SCIENCE

西北工业大学 | Northwestern Polytechnical University

数据库系统原理 | Database System

第二章 关系数据库

赵晓南

2024.09

课前复习

第二章：关系模型

- 数据结构：笛卡尔积的子集（保留有意义记录）
- 完整性约束：实体完整性、参照完整性、用户自定义完整性
- 候选码、主码、超码、主/非主属性
- 数据存储：文件
- 数据操作：关系代数
  - 集合运算（并、交、差、笛卡尔积）
  - 专门运算（选择、投影、连接、除法）
  - 比较运算
  - 逻辑运算

A	B	C
a1	b1	C1
a1	b2	C2
a2	b2	c1

$\sigma_{B=b1 \wedge C=c1}(R)$

A	B	C
a1	b1	C1

A	B	C
a1	b1	C1
a1	b2	C2
a2	b2	c1

$\pi_{A,C}$

A	C
a1	C1
a1	C2
a2	c1

其中：并，差，笛卡尔积，投影，选择 是5种基本运算。

课前复习 - 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

连接中舍弃的元组（悬浮元组），

进行保留的连接称做外连接

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

单选题 1分

设置

学生(学号,姓名,性别,专业号,年龄)

课程(课程号,课程名,学分)

选修(学号,课程号,成绩)

关于 以上选修关系描述错误的是：

A 选修关系有两个外码，分别是学号，课程号

B （学号，课程号）的组合是选修关系的主码

C （学号，课程号）的组合是选修关系的主属性

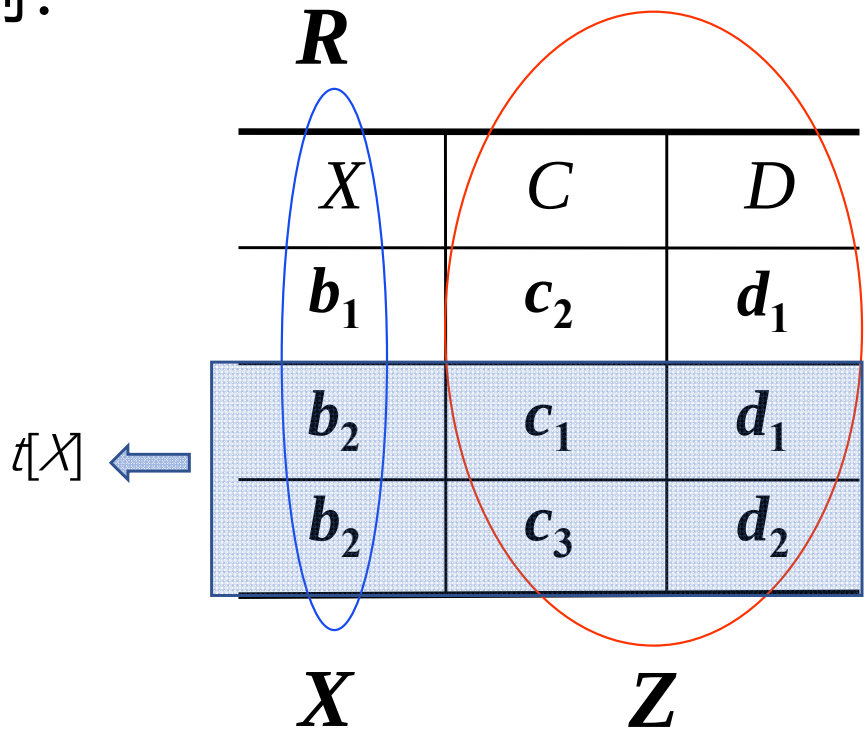
D 学号、课程号分别是选修关系的主属性，成绩是非主属性

提交

本章目录

2.4 关系代数 - 除法(象集)

- 象集Z：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：Z_x ={ t[Z] | t ∈R，t[X] =x }，它表示R 中属性组X 上值为x的诸元组在属性组Z上分量的集合。
- 例：



$Z_{X=b2}$

C	D
c_1	d_1
c_3	d_2

2.4 关系代数 - 除法

除(Division)

R[X,Y] , S[Y, Z]

- $R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_y(S) \subseteq Y_x\}$
- Y_x ： X分量x在R上的**象集**
- $\pi_y(S)$ ： S在Y上的投影

R

X	Y
x1	y1
x1	y2
x1	y3

S

Y	C	D
y1	d1	f1
y2	d2	f2

$\pi_{y(S)}$

Y
y1
y2

$\subseteq$

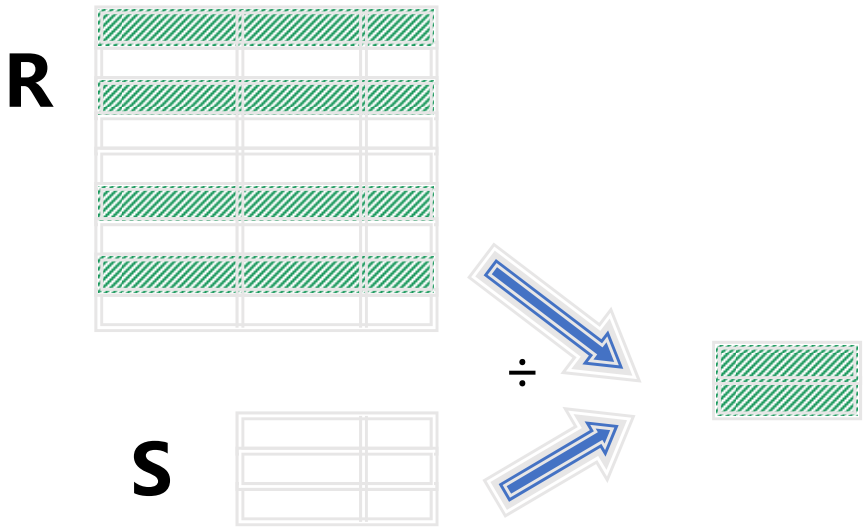
Y_{x1}

Y
y1
y2
y3

$R \div S$

X
x_1

2.4 关系代数 - 除法



计算步骤：

1. 确定结果集的属性集
2. 确定每个分量的象集Y
3. 计算S在属性集Y上的投影
4. 将满足包含条件的分量加入结果集中

除操作是同时从行和列角度进行运算

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

除法例题：例6

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

$(B,C)_{a1}$

$(B,C)_{a2}$

$(B,C)_{a3}$

$(B,C)_{a4}$

$\supseteq$

S

B	C	D
b_1	C_2	d_1
b_2	C_1	d_1
b_2	C_3	d_2

$\pi_{B,C}(S)$

$R \div S$

A
a_1

2.4 关系代数 - 除法 - 例题

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

思考：除法的用处?

student

Sno	Sname	Sgender
2001	张敏	女
2002	李明	女
2003	王亮	男

contest

cno	cname
1	数模竞赛
2	ACM竞赛
3	英语竞赛

K

Cno
1
2

award

Sno	cno	level	grade
2001	1	国际级	一等奖
2001	2	全国级	二等奖
2002	2	省级	二等奖
2003	3	校级	一等奖
2003	3	全国级	二等奖

例5：查询至少获得了数模竞赛和ACM竞赛奖励的学生学号。

$\pi_{Sno,Cno}(award) \div K$

重命名：ρ （参考英文版教材）

$\rho_s(R)$表示把关系R重命名为S，若在改名的同时还需要重命名列的名字，则表示为$\rho_s(A1,A2,...,An)(R)$，即R的各列名称为Ai。

例子：

1. 将Student表更名为S

$$\rho_s(Student)$$

2. 假设Student表仅有两列Sno，Sname，要求将Student表的列更名为C1, C2

$$\rho_{C1,C2}(Student)$$

扩展关系代数：支持Aggregation
（参考6.1.4 Extended Relational-Algebra Operations）

$$\mathcal{G}_{\mathbf{sum}(salary)}(instructor)$$

The symbol $\mathcal{G}$ is the letter G in calligraphic font; read it as "calligraphic G." The relational-algebra operation $\mathcal{G}$ signifies that aggregation is to be applied, and its subscript specifies the aggregate operation to be applied. The result of the expression above is a relation with a single attribute, containing a single row with a numerical value corresponding to the sum of the salaries of all instructors.

支持的聚集函数：

- sum(): 求和
- count(): 求个数
- count-distinct()：去重后求个数
- average()：求平均值

2.4 关系代数 - 扩展 - 聚集操作例

account

branch-name	account-number	balance
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

按照branch-name将关系account分组

$$\text{branch-name}\ g\ \text{sum(balance)}\ (account)$$

branch-name	sum-balance
Perryridge	1300
Brighton	1500
Redwood	700

2.4 关系代数 - 综合练习

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

综合运算例题（本小节的所有查询均用到以下关系实例）

Student

Sno	Sname	Sgender	Smajor	Sbirthdate
801	张三	女	01	1999-12-31
802	李四	男	01	2000-05-05
803	王五	男	01	2002-10-31
804	赵六	女	02	2001-07-11
805	钱七	男	02	1999-09-15

SC

sno	cno	grade
801	C4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

Course

cno	cname	credit	cpno
C1	数据库	3.5	C2
C2	数据结构	4	C4
C3	编译原理	4	C6
C4	C语言	3	

关系模式：
Student(sno, sname, Sgender, smajor, sbirthdate)
Course(cno, cname, credit, cpno)
SC(sno, cno, grade)

2.4 关系代数 - 综合练习

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

Student

Sno	Sname	Sgender	Smajor	Sbirthdate
801	张三	女	01	1999-12-31
802	李四	男	01	2000-05-05
803	王五	男	01	2002-10-31
804	赵六	女	02	2001-07-11
805	钱七	男	02	1999-09-15

SC

sno	cno	grade
801	c4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

例1：查询选修了C2课程的学生学号和成绩。

$$\pi_{sno,grade}\ (\ \sigma_{cno='c2'}\ (SC)\ )$$

例2：查询选修了C2课程的学生学号和姓名。

$$\pi_{sno,sname}\ (\ \sigma_{cno='c2'}\ (\ Student \bowtie SC\ ))$$

SC

sno	cno	grade
801	c4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

例3 查询未选修C2课的学生学号。

$$\pi_{sno}\ (\sigma_{cno \neq \text{'c2'}}(SC))$$

原因：如果该学生选c2课外还选了别的课，则非c2课的记录是符合条件的，此时会将该记录的学号置入结果集。

解决方法：先求出所有选c2课的学生，再从全体学生集合中减去这些学生。

$$\pi_{sno}(Student) - \pi_{sno}(\sigma_{cno= \text{'c2'}}(SC))$$

2.4 关系代数 - 综合练习

cno1	cname1	credit1	cpno1	cno2	cname2	credit2	cpno2
C1	数据库	4	C2	C2	数据结构	4	C4
C2	数据结构	4	C4	C4	Pascal	3	
C3	编译原理	4	C6	C6	操作系统	5	C7
C4	Pascal	3					

例4 查询C1课的先修课的先修课的课程号。

C

cno	cname	credit	cpno
c_1	数据库	4	c_2
c_2	数据结构	4	c_4
c_3	编译原理	4	c_6
c_4	Pascal	3	

设：$R = \sigma_{cpno1=cno2}(CXC)$

$$\pi_{cno1,cpno2}(\sigma_{cno1='c1'}(R))$$

或者 $$\pi_{cno1,cpno2}(\sigma_{cno1='c1'}(C \underset{cpno1=cno2}{\bowtie} C))$$

例5：查询至少选修C3号课程和C4号课程的学生学号。

SC

sno	cno	grade
801	C4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

K

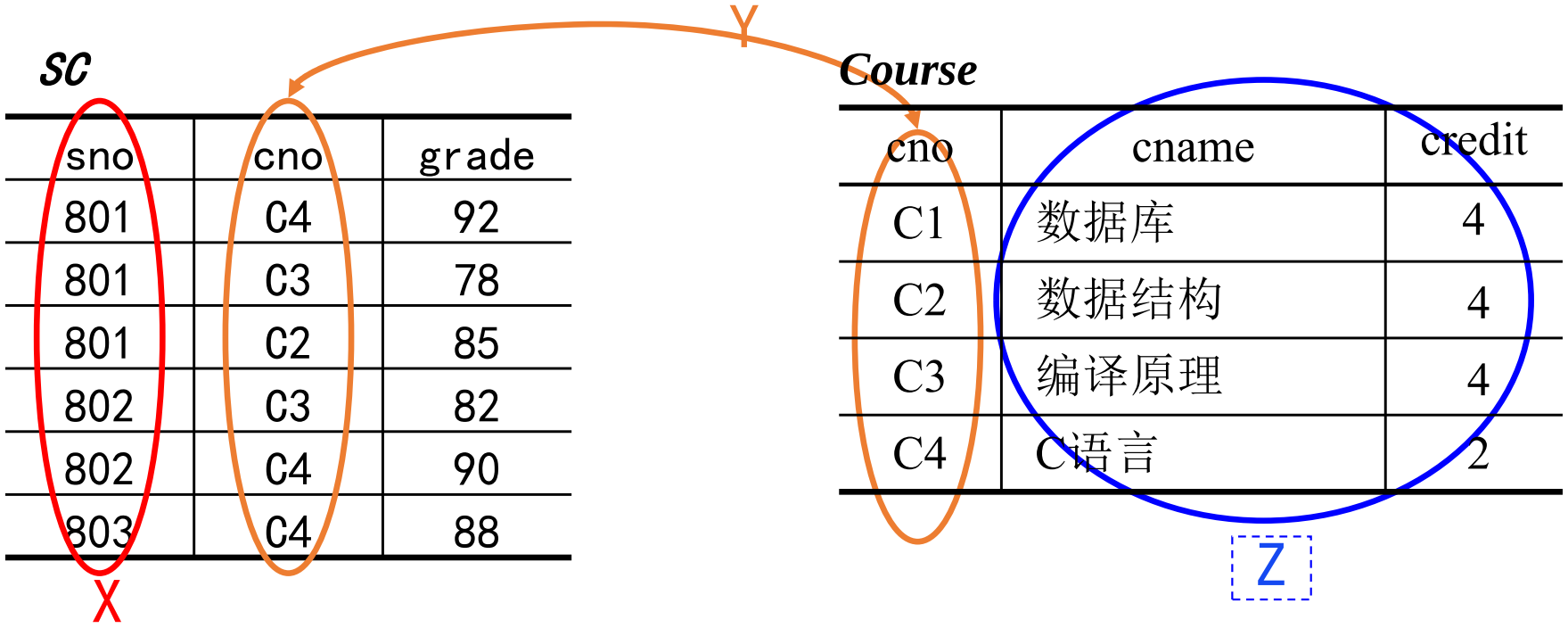
Cno
C3
C4

Sno	Cno
801	C4
801	C3
801	C2
802	C3
802	C4
803	C4

$\pi_{Sno,Cno}(SC) \div K$ ⟹

Sno
801
802

例6 查询选修全部课程的学生学号。



$$\pi_{sno,\,cno}(SC) \div \pi_{cno}(Course)$$

试做：查询至少选修了一门其直接先修课号为C5的学生姓名

$$\pi_{sname}(\sigma_{cpno='C5'}(Course) \bowtie SC \bowtie \pi_{sno,name}(Student))$$

2.4 关系代数 - 综合练习

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

Student

Sno	Sname	Sgender	Smajor	Sbirthdate
801	张三	女	01	1999-12-31
802	李四	男	01	2000-05-05
803	王五	男	01	2002-10-31
804	赵六	女	02	2001-07-11
805	钱七	男	02	1999-09-15

SC

sno	cno	grade
801	c4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

例7 查询选修了C2或C4课程的学生学号。

$$\pi_{sno}(\sigma_{cno='c2' \vee cno='c4'}(SC))$$

例8 查询只选修C2和C4课程的学生学号。

$$\pi_{sno}(\sigma_{cno='c2' \wedge cno='c4'}(SC))$$

没有哪一条记录会
在一个属性列上
取两个不同的值！

主观题 10分

设置

关系代数：查询只选修C2和C4课程的学生学号。

Student

Sno	Sname	Sgender	Smajor	Sbirthdate
801	张三	女	01	1999-12-31
802	李四	男	01	2000-05-05
803	王五	男	01	2002-10-31
804	赵六	女	02	2001-07-11
805	钱七	男	02	1999-09-15

SC

sno	cno	grade
801	c4	92
801	C3	78
801	C2	85
802	C2	82
802	C4	90
803	C4	88

作答

查询只选修C2和C4课程的学生学号。

思路

R1：至少选修了C2和C4课程的学生学号。

⇓

R2：选修了C2和C4以外其他课的学生学号。

⇓

R = R1 - R2

1）求解R1：至少选修了C2和C4课程的学生学号。

方法一：集合运算：选C2的学生与选C4的学生的交集

$$\pi_{sno}(\sigma_{cno='c2'}(SC)) \cap \pi_{sno}(\sigma_{cno='c4'}(SC))$$

方法二：除法运算：设K={C2,C4}

$$\pi_{sno,cno}(SC) \div K$$

方法三：连接运算

SC

sno	cno	grade
801	C4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

$\sigma_{1=4}(SC \times SC)$

Sc1. sno	Sc1. cno	Sc1. gradee	Sc2. sno	Sc2. cno	Sc2. gradee
801	C4	92	801	C4	92
801	C4	92	801	C3	78
801	C4	92	801	C2	85
801	C3	78	801	C4	92
801	C3	78	801	C3	78
801	C3	78	801	C2	85
801	C2	85	801	C4	92
801	C2	85	801	C3	78
801	C2	85	801	C2	85
802	C3	82	802	C3	82
..	..	..	..	..	..

$$\pi_1(\sigma_{1=4 \wedge 2='c2' \wedge 5='c4'}(SC \times SC))$$

或者 $$\pi_1(\sigma_{1=4 \wedge 5='c2' \wedge 2='c4'}(SC \times SC))$$

2）求解R2：选修了C2和C4以外其他课的学生学号。

思路1：除了C2和C4之外还选其他课的学生学号

$$\pi_{1}(\sigma_{2\neq\text{'c2'}\wedge 2\neq\text{'c4'}}(R1\bowtie SC)$$

思路2：不管是否选C2和C4，只要选了C2和C4以外的课的学生学号

方法1：

$$\pi_{1}(\sigma_{2\neq\text{'c2'}\ \wedge 2\neq c4'}(SC))$$

方法2：

设K={C2, C4}

$$\pi_{1}(\pi_{sno,cno}(SC)\bowtie(\pi_{cno}(SC)-K))$$

3）求解R：只选修了C2和C4课程的学生学号 => R1 - R2

查询只选修C2和C4课程的学生学号。　　设K={C2, C4}

$$(\pi_{sno,cno} - \pi_{sno,cno}(\sigma_{cno\neq'c2'\wedge cno\neq'c4'}(SC))) \div K$$

sno	cno	result
801	C2	√
801	C4	
802	C2	×
803	C3	×
804	C2	×
804	C4	
804	C5	

−

sno	cno
803	C3
804	C5

=

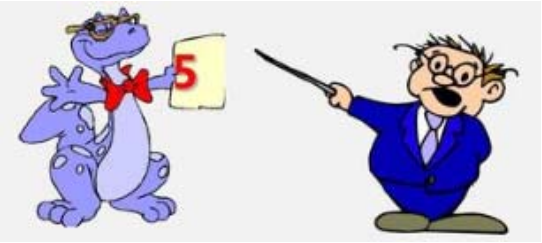
sno	cno	÷ K
801	C2	√
801	C4	
802	C2	×
804	C2	
804	C4	√ （应×）

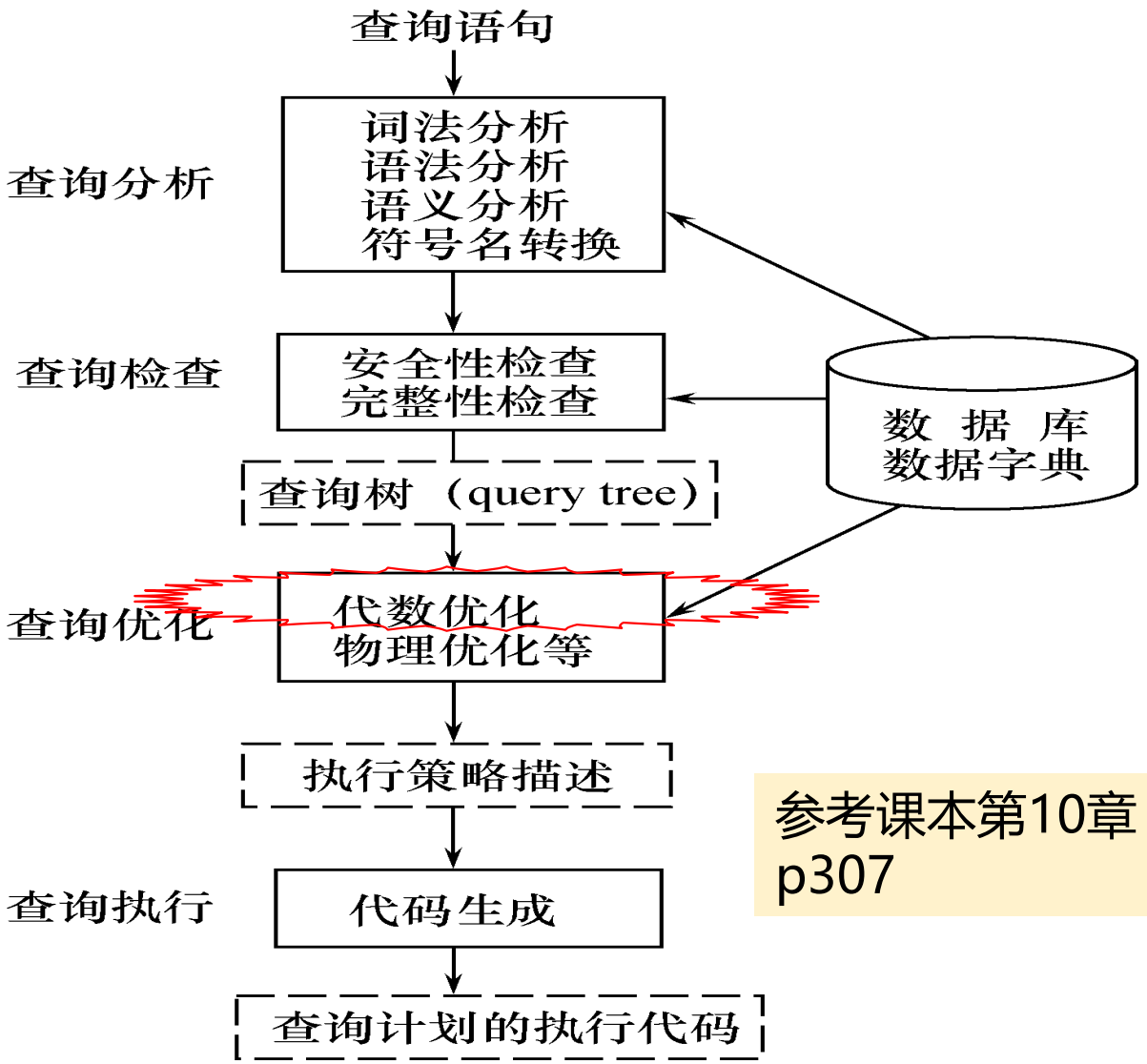
$$((\pi_{sno} - \pi_{sno}(\sigma_{cno\neq'c2'\wedge cno\neq'c4'}(SC))) \bowtie \pi_{sno,cno}(SC)) \div K$$ √

2.4 关系代数 - 作用与意义

- 建立查询思想
- 进行查询优化

Why ?





参考课本第10章 p307

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

- 查询思想与处理

查询计划

```
mysql> explain select sno, sname from s where sno=2001\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: s
   partitions: NULL
         type: ALL
possible_keys: PRIMARY
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 4
     filtered: 25.00
        Extra: Using where
1 row in set, 3 warnings (0.00 sec)
```

```
mysql> explain  format=tree select sno,sname from s where sno=2001;
+-----------------------------------------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                                                           |
+-----------------------------------------------------------------------------------------------------------------------------------+
| -> Filter: (s.sno = 2001)  (cost=1.40 rows=1)
    -> Index scan on s using IX_ngd  (cost=1.40 rows=4)
 |
+-----------------------------------------------------------------------------------------------------------------------------------+
1 row in set, 3 warnings (0.00 sec)
```


```
{
 "query_block": {
  "select_id": 1,
  "cost_info": {
    "query_cost": "1.40"
  },
  "table": {
    "table_name": "s",
    "access_type": "index",
    "possible_keys": [
      "PRIMARY"
    ],
    "key": "IX_ngd",
    "used_key_parts": [
      "sname",
      "sgender",
      "sdept"
    ],
    "key_length": "97",
    "rows_examined_per_scan": 4,
    "rows_produced_per_join": 1,
    "filtered": "25.00",
    "using_index": true,
    "cost_info": {
      "read_cost": "1.30",
      "eval_cost": "0.10",
      "prefix_cost": "1.40",
      "data_read_per_join": "128"
    },
    "used_columns": [
      "sno",
      "sname"
    ],
    "attached_condition": "(`student`.`s`.`sno` = 2001)"
  }
 }
}
```

查询优化

- RDBMS通过某种代价模型计算出各种查询执行策略的执行代价，然后选取代价最小的执行方案。

- **集中式数据库**
  - 执行开销主要包括：
    - 磁盘存取块数(I/O代价)
    - 处理机时间(CPU代价)
    - 查询的内存开销
  - I/O代价是最主要的

- **分布式数据库**
  - 总代价=I/O代价+CPU代价+内存代价 + 通信代价

- 例子：求选修了2号课程的学生姓名。

假定数据库中有1000个学生记录, 10000个选课记录，其中选修2号课程的选课记录为50个

$$Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno\wedge Sc.Cno='2'}(Student\times SC))$$

$$Q_2=\pi_{Sname}(\sigma_{Sc.Cno='2'}(Student\bowtie SC))$$

$$Q_3=\pi_{Sname}(Student\bowtie \sigma_{Sc.Cno='2'}(SC))$$

见下页

求选修了2号课程的学生姓名，Q1,Q2,Q3谁的效率高?

假定数据库中有1000个学生记录, 10000个选课记录，其中选修2号课程的选课记录为50个.

A: $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno\wedge Sc.Cno='2'}(Student\times SC))$

B: $Q_2=\pi_{Sname}(\sigma_{Sc.Cno='2'}(Student \bowtie SC))$

C: $Q_3=\pi_{Sname}(Student \bowtie \sigma_{Sc.Cno='2'}(SC))$

A B C

提交

设置

求选修了2号课程的学生姓名，效率最高方法是效率最低方法的多少倍?

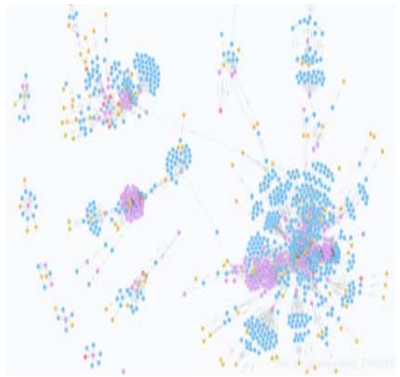
假定数据库中有1000个学生记录, 10000个选课记录, 其中选修2号课程的选课记录为50个.

A: $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno\wedge Sc.Cno='2'}(Student\times SC))$

B: $Q_2=\pi_{Sname}(\sigma_{Sc.Cno='2'}(Student \bowtie SC))$

C: $Q_3=\pi_{Sname}(Student \bowtie \sigma_{Sc.Cno='2'}(SC))$

A 100倍　B 1000倍　C 10000倍



提交

2.4 关系代数 - Q1查询效率分析

- $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno\wedge Sc.Cno='2'}(Student\times SC))$

假定数据库中有1000个学生记录, 10000个选课记录，其中选修2号课程的选课记录为50个.

整体步骤：

1. 笛卡尔积运算
   1) 读Student表和SC表
   2) 两表记录组合输出到中间文件

2. 选择运算
   1) 读笛卡尔积输出的中间文件，边读边判断是否符合选择的条件（Student.Sno=SC.Sno∧Sc.Cno='2' ）

3. 结果列投影的运算

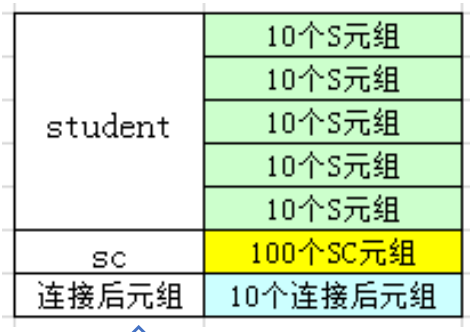
2.4 关系代数 - Q1查询效率分析

- $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno \wedge Sc.Cno='2'}(Student \times SC))$

假定数据库中有1000个学生记录，10000个选课记录，其中选修2号课程的选课记录为50个.

1. 笛卡尔积 – ① 读两个表：S和SC

读总块数为 2100块
=100+2000



$$\frac{1000}{10}+\frac{1000}{10\times 5}\times\frac{10000}{100}$$

读S表：一共需读S表100块
(1000/10=100)

笛卡尔积：
读SC表20遍，每遍100块(次)

假设：
1. 内存中的一个块可以存储10个Student元组或100个SC元组或10个连接后元组。
2. 内存中可以放5块Student元组和1块SC元组。

1. 第1次读S表S1-S5块，第1次读SC表SC1块，做笛卡尔积
2. S表S1-S5保持不动，第2次读SC表SC2块，做笛卡尔积
   ...（读SC表共重复 10000/100=100次，每次1块）
3. 第2次读S表S6-S10块，第1次读SC表SC1块，做笛卡尔积
   ...（读SC表共重复 10000/100=100次，每次1块）
4. 第20次读S表S96-S100块，第1次读SC表SC1块，做笛卡尔积
   （ 1000/(5*10)=20 ）

所以笛卡尔积，需要读SC表一共：20*100 = 2000块

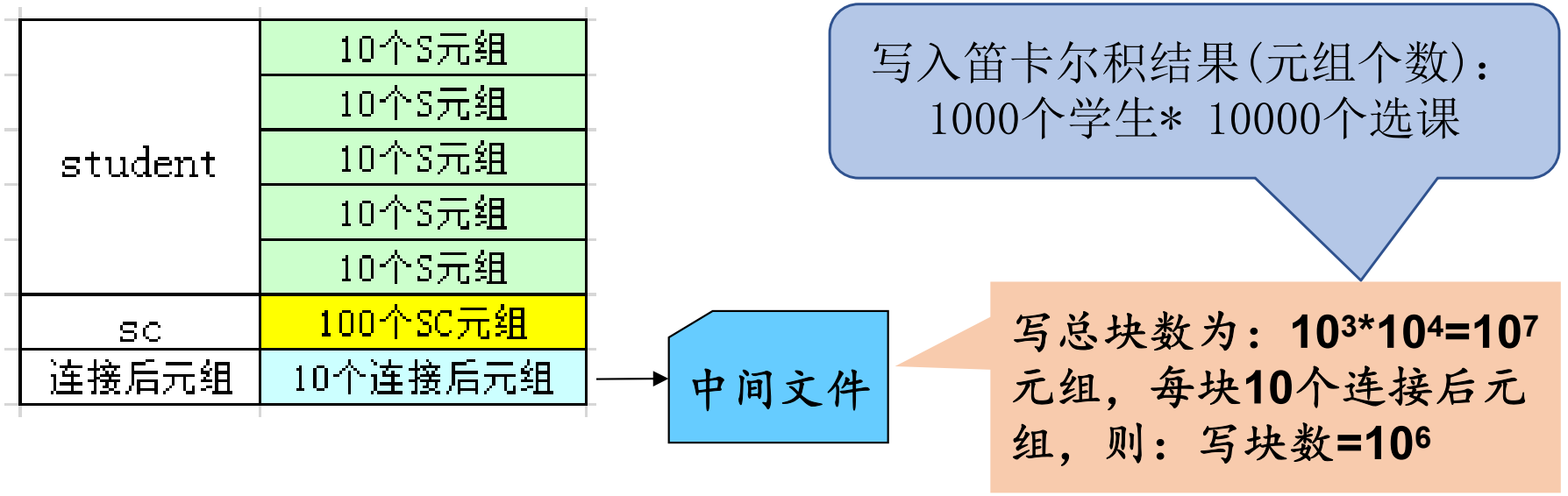
2.4 关系代数 - Q1查询效率分析

- $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno \wedge Sc.Cno='2'}(Student \times SC))$

假定数据库中有1000个学生记录，10000个选课记录，其中选修2号课程的选课记录为50个.

1. 笛卡尔积（②笛卡尔积的运算结果写到中间文件）

设内存中的一个块可以存储10个S元组或100个SC元组或10个连接后元组。内存中可以放5块Student元组和1块SC元组。

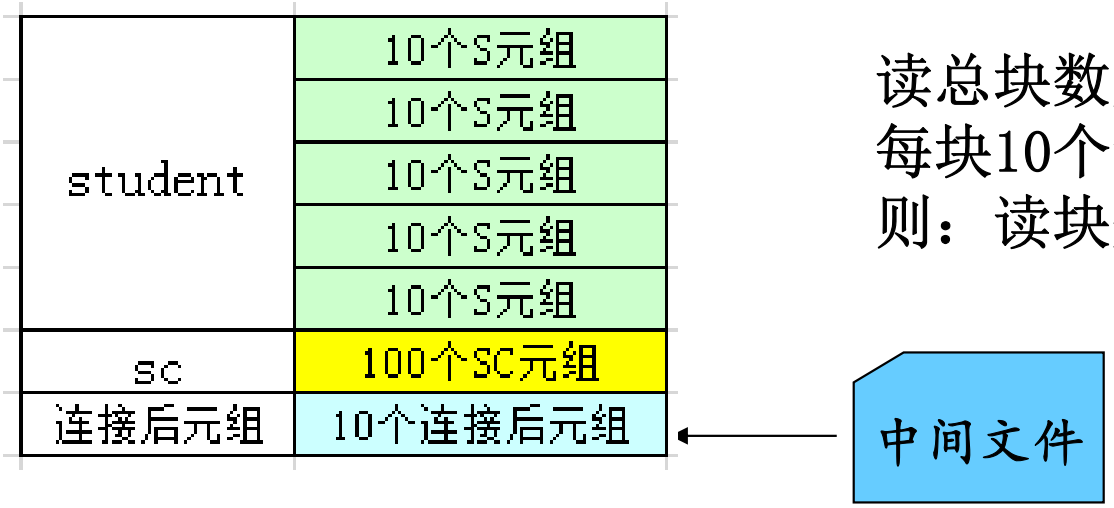


- $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno \wedge Sc.Cno='2'}(Student \times SC))$

假定数据库中有1000个学生记录，10000个选课记录，其中选修2号课程的选课记录为50个

2. 选择操作（主要考虑读操作，条件判断时间可忽略）

从中间文件读取连接后的元组，读取块数 10^6



读总块数为$10^3*10^4=10^7$块
每块10个连接后元组
则：读块数=10^6

- $Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno\wedge Sc.Cno='2'}(Student\times SC))$

假定数据库中有1000个学生记录，10000个选课记录，其中选修2号课程的选课记录为50个

3. 投影操作

选择操作的结果在sname上投影，即可得到结果。

第一种情况的总计执行查询，假设每秒读写20块

读写块数： $2100 + 10^6 + 10^6$

读写秒数：读写块数/20 = $105 + 1\times10^5\approx10^5$ s

- 所有内存处理时间均忽略不计

2.4 关系代数 - Q2查询效率分析

- $Q_2=\pi_{Sname}(\sigma_{Sc.Cno='2'}(Student \bowtie SC))$

假定数据库中有1000个学生记录，10000个选课记录，其中选修2号课程的选课记录为50个

1. 计算自然连接
   - 执行自然连接，读取Student和SC表的策略不变，总的读取块数仍为2100块花费105s。
   - 自然连接的结果比第一种情况大大减少=>读写中间文件小
     自然连接：10^4个（以数据量大的学生选课表为驱动表）
     写出这些元组时间为$10^4/10/20$=50s。
2. 读取中间文件块，执行选择运算。花费时间也为50s。
3. 把第2步结果投影输出。

第二种情况：总的执行时间≈105+50+50≈205s
（最后一步：选择2号记录仅50个，处理开销小可以忽略）

2.4 关系代数 - Q3查询效率分析

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

- $Q_3=\pi_{Sname}(\text{Student} \bowtie \sigma_{Sc.Cno='2'}(SC))$

假定数据库中有1000个学生记录，10000个选课记录，其中选修2号课程的选课记录为50个

1. 先对SC表作选择运算，只需读一遍SC表，存取100块花费时间为5s，因为满足条件的元组仅50个，不必使用中间文件保存这个临时结果。

2. 读取Student表，把读入的Student元组和内存中的SC元组作连接。也只需读一遍Student表共100块，花费时间为5s。

3. 把连接结果投影输出

第三种情况总的执行时间≈5+5≈10s

- 时间代价差异巨大

$Q_1=\pi_{Sname}(\sigma_{Student.Sno=SC.Sno \wedge Sc.Cno='2'}(Student \times SC))$ 10^5s

$Q_2=\pi_{Sname}(\sigma_{Sc.Cno='2'}(Student \bowtie SC))$ 205s

$Q_3=\pi_{Sname}(Student \bowtie \sigma_{Sc.Cno='2'}(SC))$ 10s

效率最高方法是效率最低方法的多少倍？

A. 100倍， B：1000倍， C:10000倍

2.4 关系代数 - 小结

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

关系代数总结

关系代数是在关系上定义一组运算，由已知关系经过有限次地运算可以得到目标关系(查询)。

- **传统的集合运算：**
  1. 并(Union)
  2. 交(Intersection)
  3. 差(Difference)
  4. 广义笛卡尔积 (Extended Cartesian Product)

- **专门的关系运算：**
  1. 选择(Select)
  2. 投影(Project)
  3. 连接(Join)
  4. 除(Divide)
  5. 重命名(Rename)
  6. 聚集操作

本章目录

2.5 关系演算

- 以数理逻辑中的谓词演算为基础
- 分类：按谓词变元不同分类

1.元组关系演算：

以元组变量作为谓词变元的基本对象

元组关系演算语言ALPHA

2.域关系演算：

以域变量作为谓词变元的基本对象

域关系演算语言QBE

2.5.1 元组关系演算---ALPHA语言

检索语句：GET

更新语句：PUT，HOLD，UPDATE，DELETE，DROP

语句的基本格式：

操作语句　工作空间名（表达式）：　操作条件

表达式：指定语句的操作对象。

关系名|关系名.属性名

操作条件：将操作结果限定在满足条件的元组中，

格式为逻辑表达式。

2.5.1 元组关系演算---ALPHA语言

SC

sno	cno	grade
801	C4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

一、检索操作

(1)简单检索(即不带条件的检索)

GET 工作空间名 (表达式)

例1 查询所有被选修的课程号码 GET W (SC.Cno)

例2 查询所有学生的数据 GET W (Student)

(2) 限定的检索(即带条件的检索)

GET 工作空间名 (表达式)：操作条件

例3 查询计算机专业(CS)2000年以前出生的学生的学号和出生日期

GET W (Student.Sno，Student.Sbirthdate):

Student.Smajor='CS'∧ Student.Sbirthdate<2000-1-1

$$\pi_{Sno,\ Sbirthdate}(\sigma_{Sbirthdate<2000\text{-}1\text{-}1 \wedge Smajor=\text{“CS”}}(Student))$$

2.5.1 元组关系演算---ALPHA语言

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

(3) 带排序的检索

GET 工作空间名 (表达式1) [：操作条件] DOWN/UP 表达式2

例4 查询计算机专业(CS)学生的学号、出生日期，结果按年龄降序排序

GET W (Student.Sno，Student.Sbirthdate):

Student.Smajor='CS' DOWN Student.Sbirthdate

(4) 带定额的检索

GET 工作空间名 (定额) (表达式1) [：操作条件]

[DOWN/UP 表达式2]

例5 查询选修了81003课程、成绩在前三的学生的学号及其成绩。

GET W (3) (SC.Sno, SC.Grade,):

SC.Cno= '81003' DOWN SC.Grade

(5)用元组变量的检索

- 元组变量是可以在某一关系范围内变化的
  (也称为范围变量Range Variable)

- 元组变量的用途
  ① 简化关系名：设一个较短名字的元组变量代替较长名
  ② 操作条件中使用量词时必须用元组变量

- 定义元组变量
  RANGE 关系名 变量名

  例6 查询信息安全专业学生的名字

  RANGE Student X
  GET W (X.Sname): X.Smajor= 'IS'

(6) 用存在量词的检索

∃:存在量词

例7 查询选修2号课程的学生名字

RANGE SC X

GET W (Student.Sname):

∃X(X.Sno=Student.Sno ∧ X.Cno='2')

(7) 带有多个关系的表达式的检索

例8 查询成绩为90分以上的学生名字与课程名字

RANGE SC SCX

GET W (Student.Sname, Course.Cname):

∃SCX (SCX.Grade ≥ 90 ∧

SCX.Sno=Student.Sno ∧ SCX.Cno=Course.Cno)

2.5.1 元组关系演算---ALPHA语言

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

SC

sno	cno	grade
801	C4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

(8)用全称量词的检索

例9 查询不选C1号课程的学生名字

RANGE SC SCX

GET W (Student.Sname):

∀ SCX (SCX.Sno≠Student.Sno∨SCX.Cno≠'C1')

或者 ¬∃SCX (SCX.Sno=Student.Sno∧SCX.Cno='C1')

(9) 用两种量词的检索

例10 查询选修了全部课程的学生姓名

RANGE SC SCX, Course CX

GET W (Student.Sname):

∀CX ∃SCX (SCX.Sno=Student.Sno∧

SCX.Cno=CX.Cno)

2.5.1 元组关系演算---ALPHA语言

(10) 用蕴涵的检索

P:1002选了该课程

Q:某学生也选了该课程

P	Q	P->Q
0	0	1
0	1	1
1	0	0
1	1	1

例11 查询最少选修了1002学生所选课程的学生学号。

RANGE Course CX

SC SCX

SC SCY

⟹ GET W (Student.Sno):

∀CX(∃SCX(SCX.Sno='1002'∧SCX.Cno=CX.Cno)

∃SCY(SCY.Sno=Student.Sno∧SCY.Cno= CX.Cno))

(11) 集函数

常用集函数(Aggregation function)

COUNT：对元组计数

TOTAL：求总和

MAX：求最大值

MIN：求最小值

AVG：求平均值

例12 查询学生所在系的数目

GET W (COUNT(Student.Sdept))

课本例题对比：

例2.22：查询选修直接先行课是6号课程的学生学号。

RANGE Course CX

GET W(SC.sno):∃CX(CX.Cno=SC.Cno)∧(CX.Pcno='6')

例2.23：查询至少选修一门其直接先行课为6号课程的学生姓名。

RANGE Course CX

SC SCX

GET W(Student.sname):

∃SCX(SCX.Sno=Student.sno)∧

∃CX(CX.Cno=SCX.Cno)∧(CX.Pcno='6')

2.5.1 元组关系演算---ALPHA语言

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

1. 修改操作 HOLD - MOVE - UPDATE

步骤:①用HOLD语句将要修改的元组从数据库中读到工作空间中

HOLD 工作空间名（表达式）[：操作条件]

HOLD语句是带上并发控制的GET语句

② 用宿主语言修改工作空间中元组的属性

③ 用UPDATE语句将修改后的元组送回数据库中

UPDATE 工作空间名

例1: 把1007学生从计算机科学系转到信息系(p68)

HOLD W (Student.Sno, Student.Sdetp): Student.Sno='1007'

(从Student关系中读出1007学生的数据)

MOVE 'IS' TO W.Sdept （用宿主语言进行修改）

UPDATE W （把修改后的元组送回Student关系）

2. 插入操作 PUT

① 用宿主语言在工作空间中建立新元组

② 用PUT语句把该元组存入指定关系中

PUT 工作空间名 （关系名）

PUT语句只对一个关系操作

例2: 学校新开设了一门2学分的课程“计算机组织与结构”，其课程号为8，直接先行课为6号课程。插入该课程元组

```
MOVE '8' TO W.Cno
MOVE '计算机组织与结构' TO W.Cname
MOVE '6' TO W.Cpno
MOVE '2' TO W.Ccredit
PUT W (Course)
```

3. 删除 DELETE

步骤：

①用HOLD语句把要删除的元组从数据库中读到工作空间中

②用DELETE语句删除该元组

DELETE 工作空间名

例3:1010学生因故退学，删除该学生元组。

HOLD W (Student): Student.Sno='1010'

DELETE W

2.5.1 元组关系演算---ALPHA语言

检索操作-- GET

GET 工作空间名 [（定额）] (表达式1)

[: 操作条件] [DOWN/UP 表达式2]

插入操作

MOVE--PUT

修改操作

HOLD - MOVE - UPDATE

删除操作

HOLD--DELETE

本节课重点

- 关系代数

  除，关系代数的综合运算

- 关系演算语言(ALPHA)

作业：

1. 第二章课后：第6,8题（仅用关系代数、ALPHA语言）
2. 补充作业（见PPT下一页，6.14, 6.15）

补充作业2 (用关系代数写表达式)

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

6.14 Consider the following relational schema for a library:

member(*memb_no*, *name*, *dob*)
books(*isbn*, *title*, *authors*, *publisher*)
borrowed(*memb_no*, *isbn*, *date*)

dob: date of birth

Write the following queries in relational algebra.

a. Find the names of members who have borrowed any book published by "McGraw-Hill".

b. Find the name of members who have borrowed all books published by "McGraw-Hill".

c. Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".

d. For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.

e. Find the average number of books borrowed per member. Take into account that if an member does not borrow any books, then that member does not appear in the *borrowed* relation at all.

employee (*person_name*, *street*, *city*)
works (*person_name*, *company_name*, *salary*)
company (*company_name*, *city*)
manages (*person_name*, *manager_name*)

用关系代数写表达式

6.15 Consider the employee database of Figure 6.22. Give expressions in tuple relational calculus and domain relational calculus for each of the following queries:

a. Find the names of all employees who work for "First Bank Corporation".

b. Find the names and cities of residence of all employees who work for "First Bank Corporation".

c. Find the names, street addresses, and cities of residence of all employees who work for "First Bank Corporation" and earn more than $10,000.

d. Find all employees who live in the same city as that in which the company for which they work is located.

e. Find all employees who live in the same city and on the same street as their managers.

f. Find all employees in the database who do not work for "First Bank Corporation".

g. Find all employees who earn more than every employee of "Small Bank Corporation".

h. Assume that the companies may be located in several cities. Find all companies located in every city in which "Small Bank Corporation" is located.