

数 据 管 理 基 础

第2章 关系数据库

(关系代数的应用)

智能软件与工程学院



关系代数的应用

- ❑ MyCAP数据库
- ❑ 基本运算符的查询表示
- ❑ 扩充运算符的查询表示
- ❑ 小结

MyCAP数据库 1

□ 设有一个公司产品零售数据库，其关系模式如下(编号属性分别是各个关系的码)：

➤ 顾客关系(顾客编号, 姓名, 居住城市, 折扣)

C(cid, cn, city, dis)

➤ 供应商关系(供应商编号, 名称, 所在城市, 佣金比例)

A (aid, an, city, per)

➤ 商品关系(商品编号, 名称, 库存城市, 库存数量, 单价)

P (pid, pn, city, qua, pri)

➤ 订单关系(订单编号, 订购日期, 顾客编号, 供应商编号, 商品编号, 订购数量, 销售金额)

O(ordno, orddate, cid, aid, pid, qty, dols)

MyCAP数据库 2

顾客关系 **C(cid, cn, city, dis)**

<u>cid</u>	cname	city	disct
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

MyCAP数据库 3

供应商关系 **A (aid, an, city, per)**

<u>aid</u>	aname	city	percent
a01	Smith	New York	6
a02	Jones	Newark	6
a03	Brown	Tokyo	7
a04	Gray	New York	6
a05	Otasi	Duluth	5
a06	Smith	Dallas	5

MyCAP数据库 4

商品关系 P (pid, pn, city, qua, pri)

<u>pid</u>	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
p06	folder	Dallas	123100	2.00
p07	case	Newark	100500	1.00

MyCAP数据库 5

订单关系 **O(ordno, orddate, cid, aid, pid, qty, dols)**

<u>ordno</u>	orddate	cid	aid	pid	qty	dollars
1011	jan	c001	a01	p01	1000	450.00
1012	jan	c001	a01	p01	1000	450.00
1019	feb	c001	a02	p02	400	180.00
1017	feb	c001	a06	p03	600	540.00
1018	feb	c001	a03	p04	600	540.00
1023	mar	c001	a04	p05	500	450.00
...

□ 求解过程

- 1) 确定查询目标（确定结果关系的关系模式）
- 2) 明确查询条件（确定结果元组的语义）
- 3) 选择从条件到目标的查找路径，并据此确定操作对象，即：
 - 在操作过程中需要使用到那些关系？
 - 这些关系又是如何联合起来实现查询的？
- 4) 根据以上分析构造关系代数表达式

关系代数的应用

- ❑ MyCAP数据库
- ❑ 基本运算符的查询表示
- ❑ 扩充运算符的查询表示
- ❑ 小结

基本运算符的查询表示

1. 查询所有顾客的姓名；
2. 查询所有顾客所居住的城市名称；
3. 查询所有居住在京都的顾客；
4. 查询存放于达拉斯且单价超过\$0.50的商品；
5. 查询所有折扣低于10%的顾客所在城市以及佣金低于6%的代理商所在城市的城市名称；
6. 查询位于同一个城市的顾客和代理商，给出顾客的姓名、代理商的名称以及他们所在城市的名称；
7. 查询姓名为‘Allied’的顾客的订购信息，列出其每一条订单记录所购买商品的商品编号、订购日期以及订购数量；
8. 查询顾客、代理商以及所订购的商品都位于同一个城市的订单的编号；
9. 在当前的所有顾客中，查询折扣最高的顾客的编号。

Example 1

□ 查询所有顾客的姓名

令结果关系为CN，则： $CN := \pi_{cname}(C)$

C

<u>cid</u>	cname	city	discent
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

CN

cname
TipTop
Basics
Allied
ACME

Example 2

❑ 查询所有顾客所居住的城市名称

C

<u>cid</u>	cname	city	discent
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

$\pi_{\text{city}}(\mathbf{C})$

city
Duluth
Dallas
Kyoto

❑ 要在结果关系中剔除重复出现的‘Duluth’及‘Dallas’元组

Example 3

□ 查询所有居住在京都(Kyoto)的顾客

CUSTOMERS

<u>cid</u>	cname	city	discent
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

$T := \sigma_{\text{city} = \text{'Kyoto'}} (\text{CUSTOMERS})$

<u>cid</u>	cname	city	discent
c006	ACME	Kyoto	0.00

Example 4

❑ 查询存放于达拉斯(Dallas)且单价(price)超过\$0.50的商品

<u>pid</u>	pname	city	quantity	price
P01	comb	Dallas	111400	0.50
p04	pen	Duluth	125300	1.00
p05	pencil	Dallas	221400	1.00
p06	folder	Dallas	123100	2.00
p07	case	Newark	100500	1.00

$T := \sigma_{\text{city} = \text{'Dallas'} \wedge \text{price} > 0.50} (\text{Products})$

Example 5

C(cno, cname, city, discent)

A(aid, aname, city, percent)

P(pid, pname, city, quantity, price)

O(ordno, orddate, cid, aid, pid, qty, dollars)

- ❑ 查询所有折扣(discent)低于10%的顾客所在城市以及佣金(percent)低于6%的代理商所在城市的城市名称

$$\pi_{\text{city}} (\sigma_{\text{discent} < 10} (C)) \cup \pi_{\text{city}} (\sigma_{\text{percent} < 6} (A))$$

【思考】该查询能不能如下表示？

$$\pi_{\text{city}} ((\sigma_{\text{discent} < 10} (C)) \cup (\sigma_{\text{percent} < 6} (A)))$$

Example 6

C(cno, cname, city, discent)

A(aid, aname, city, percent)

P(pid, pname, city, quantity, price)

O(ordno, orddate, cid, aid, pid, qty, dollars)

- ❑ 查询位于同一个城市的顾客(customers)和代理商(agents), 给出顾客的姓名(cname)、代理商的名称(aname)以及他们所在城市的名称(city)

$\pi_{C.cname, C.city, A.aname} (\sigma_{C.city = A.city} (C \times A))$

Example 7

C(cno, cname, city, discent)

A(aid, aname, city, percent)

P(pid, pname, city, quantity, price)

O(ordno, orddate, cid, aid, pid, qty, dollars)

❑ 查询姓名为‘Allied’的顾客的订购信息(orders), 列出其每一条订单记录所购买商品的商品编号(pid)、订购日期(orddate)以及订购数量(qty)

➤ 参考答案1:

$\pi_{O.pid, O.orddate, O.qty}(\sigma_{C.cid=O.cid \wedge C.cname='Allied'}(C \times O))$

➤ 参考答案2:

$\pi_{O.pid, O.orddate, O.qty}(\sigma_{C.cid=O.cid}(\pi_{cid}(\sigma_{C.cname='Allied'}(C)) \times O))$

Example 8

C(cno, cname, city, discent)

A(aid, aname, city, percent)

P(pid, pname, city, quantity, price)

O(ordno, orddate, cid, aid, pid, qty, dollars)

❑ 查询顾客、代理商以及所订购的商品都位于同一个城市的订单的编号(ordno)

$$\pi_{\text{ordno}} \left(\begin{array}{c} \delta_{\text{C.city=A.city} \wedge \text{P.city=A.city} \wedge \text{C.cid=O.cid} \wedge \text{A.aid=O.aid} \wedge \text{P.pid=O.pid}} \left(\right. \\ \quad \pi_{\text{cid,city}}(\text{C}) \times \pi_{\text{aid,city}}(\text{A}) \times \pi_{\text{pid,city}}(\text{P}) \times \text{O} \\ \left. \right) \end{array} \right)$$

Example 9

C(cno, cname, city, discent)

A(aid, aname, city, percent)

P(pid, pname, city, quantity, price)

O(ordno, orddate, cid, aid, pid, qty, dollars)

【思考】 在当前的所有顾客中, 查询折扣(discent)最高的顾客的编号(cid)

<u>cid</u>	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

❑ 在当前的所有顾客中, 查询折扣(discnt)最高的顾客的编号(cid)

➤ 错误答案:

$$\pi_{C.cid} (\sigma_{discnt = MAX(discnt)} (C))$$

<u>cid</u>	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

□ 对于一个客户c，如何判定c的折扣是“最高” or “不是最高”？

- a) 如果c的折扣不小于所有客户的折扣，则c的折扣就是最高的；
- b) 如果能够找到一个客户g，g的折扣比c的折扣高，那么c的折扣就不是最高的。

<u>cid</u>	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

□ 给定一个客户c，如何判定c的折扣“不是最高”？

b) 如果能够找到一个或一些客户g，g的折扣比c的折扣要高，那么c的折扣就不是最高的。

□ 换句话说，给定一个客户c，用条件 **c.discnt < g.discnt** 在顾客关系上进行元组选择，查找满足上述要求的顾客g。如果结果集不为空，则c就是折扣不是最高的客户。

<u>cid</u>	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

❑ 为了查询得到折扣不是最高的顾客，我们需要两个顾客集合（完全一样的关系C和关系S）

➤ 客户关系C: 作为目标对象的候选集

➤ 客户关系S: 作为条件对象的集合

❑ 通过这两个客户关系，如何在C中找出折扣不是最高的客户元组？

<u>cid</u>	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

□ 令 $S := C$

$$R_2 := \pi_{C.cid} (\sigma_{C.discnt < S.discnt} (C \times S))$$

□ R_2 即为折扣不是最高的客户编号的集合。

CUSTOMERS

<u>cid</u>	cname	city	discnt
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

- ❑ 为了后面表示上的方便，我们仅保留该关系中的cid和discnt两个属性，以及c001,c002,c003三个元组。从而得到一个简化的关系C和关系S，并演示如何去查询得到‘折扣最高’的客户。

C

cid	discnt
c001	10
c002	12
c003	8

S

cid	discnt
c001	10
c002	12
c003	8

C × S

C.cid	C.discnt	S.cid	S.discnt
c001	10	c001	10
c001	10	c002	12
c001	10	c003	8
c002	12	c001	10
c002	12	c002	12
c002	12	c003	8
c003	8	c001	10
c003	8	c002	12
c003	8	c003	8

$\sigma_{C.\text{discnt} < S.\text{discnt}} (C \times S)$

C.cid	C.discnt	S.cid	S.discnt
c001	10	c001	10
c001	10	c002	12
c001	10	c003	8
c002	12	c001	10
c002	12	c002	12
c002	12	c003	8
c003	8	c001	10
c003	8	c002	12
c003	8	c003	8

$\sigma_{C.discount < S.discount} (C \times S)$

C.cid	C.discount	S.cid	S.discount
c001	10	c002	12
c003	8	c001	10
c003	8	c002	12

原客户关系 C

cid	discount
c001	10
c002	12
c003	8

□ 在C.cid列上留下了 “折扣不是最高” 的客户的cid

C.cid	C.discnt	S.cid	S.discnt
c001	10	c002	12
c003	8	c001	10
c003	8	c002	12

原客户关系 C

cid	discnt
c001	10
c002	12
c003	8

$$\pi_{C.cid} (\sigma_{C.discnt < S.discnt} (C \times S))$$

□ 投影得到“折扣不是最高”的客户C的 cid

原客户关系 C

cid	discnt
c001	10
c002	12
c003	8

C.cid	C.discnt	S.cid	S.discnt
c001	10	c002	12
c003	8	c001	10
c003	8	c002	12

$$\pi_{S.cid} (\sigma_{C.discnt < S.discnt} (C \times S))$$

□ 换个方式, 可投影得到“折扣不是最低”的客户S的cid

$\sigma_{C.\text{discnt} < S.\text{discnt}} (C \times S)$

C.cid	C.discnt	S.cid	S.discnt
c001	10	c001	10
c001	10	c002	12
c001	10	c003	8
c002	12	c001	10
c002	12	c002	12
c002	12	c003	8
c003	8	c001	10
c003	8	c002	12
c003	8	c003	8

注意： 在这里的选择运算中，切记不能使用“<=”！

$\sigma_{C.\text{discnt} \leq S.\text{discnt}} (C \times S)$

C.cid	C.discnt	S.cid	S.discnt
c001	10	c001	10
c001	10	c002	12
c001	10	c003	8
c002	12	c001	10
c002	12	c002	12
c002	12	c003	8
c003	8	c001	10
c003	8	c002	12
c003	8	c003	8

注意：使用了 “ \leq ” 后的查询结果！

$\sigma_{C.\text{discnt} \leq S.\text{discnt}} (C \times S)$

C.cid	C.discnt	S.cid	S.discnt
c001	10	c001	10
c001	10	c002	12
c002	12	c002	12
c003	8	c001	10
c003	8	c002	12
c003	8	c003	8

C

cid	discnt
c001	10
c002	12
c003	8

使用了“ \leq ”后，在C.cid列上含有所有客户的cid！

Example 9 (cont.)

C(cno, cname, city, discent)

A(aid, aname, city, percent)

P(pid, pname, city, quantity, price)

O(ordno, orddate, cid, aid, pid, qty, dollars)

【思考】 在当前的所有顾客中, 查询折扣(discent)最高的顾客的编号(cid)

a) 查询得到“折扣不是最高”的客户的cid:

$$\pi_{C.cid} (\sigma_{C.discent < S.discent} (C \times S))$$

b) 如何得到“折扣最高”的客户的cid?

$$\pi_{C.cid} (C) - \pi_{C.cid} (\sigma_{C.discent < S.discent} (C \times S))$$

Example 9 (cont.)

□ 在当前的所有顾客中, 查询享受最大折扣(discent)的顾客的编号(cid)

① 查询所有顾客的编号, 其结果构成关系 R_1

$$R_1 := \pi_{cid} (C)$$

② 查询折扣并非最大的顾客 (其折扣低于其他某个顾客的折扣, 或至少存在一个顾客X, 而X的折扣高于当前顾客的折扣) 的编号, 查询结果构成关系 R_2

令 $S := C$, 则:

$$R_2 := \pi_{C.cid} (\sigma_{C.discent < S.discent} (C \times S))$$

③ 利用差运算获得享受最大折扣顾客的编号:

$$T := R_1 - R_2$$

Example 9 (cont.)

- (合并上述的三个公式) 在当前的所有顾客中, 查询享受最大折扣(discont)的顾客的编号(cid) (令 $S := C$)

$$\pi_{cid} (C) - \pi_{C.cid} (\sigma_{C.discont < S.discont} (C \times S))$$

- **思考题:** 下述查询是否可以用关系代数来表示? 如果可以, 请写出对应的关系代数表达式。
- ① 查询具有最小折扣的顾客的编号;
 - ② 查询折扣并非最大的顾客的编号;
 - ③ 查询其折扣为第二大的顾客的编号;
 - ④ 查询折扣等于所有顾客折扣的平均值的顾客的编号;
 - ⑤ 查询具有最大折扣的顾客的姓名(cname)。

关系代数的应用

- ❑ MyCAP数据库
- ❑ 基本运算符的查询表示
- ❑ 扩充运算符的查询表示
- ❑ 小结

扩充运算符的查询表示

□ 例子数据库：学生选课数据库

- 学生关系：S(sno, sn, sd, sa)
- 课程关系：C(cno, cn, pno)
- 选课关系：SC(sno, cno, g)

学生 S (sno, sn, sd, sa)

课程 C (cno, cn, pno)

选课 SC (sno, cno, g)

【例1】检索课程号为C，且成绩为A的所有学生姓名

$$\pi_{sn} (\sigma_{cno=C \wedge g=A} (S \bowtie SC))$$

【例2】检索s1所修读的所有课程名及其预修课号

$$\pi_{cn, pno} (\sigma_{sno=s1} (C \bowtie SC))$$

【例3】检索年龄为23岁的学生所修读的课程名

$$\pi_{cn} (\sigma_{sa=23} (S \bowtie SC \bowtie C))$$

学生 S (sno, sn, sd, sa)

课程 C (cno, cn, pno)

选课 SC (sno, cno, g)

【例4】检索至少修读为S₅所修读的一门课的学生姓名

【分析】

1) 结果元组需要满足的条件？

修读了S₅所修读过的某一门课程

2) S₅修读了哪些课程？

3) 如何从S₅修读过的课程查出满足要求的学生元组？

例4 (cont.)

1) 查询S₅选修过的课程的课程编号

$$T_1 := \pi_{cno} (\sigma_{sno=S5} (SC))$$

2) 查询选修过T₁中的某门课程的学生们的学号

$$T_2 := \pi_{sno} (SC \bowtie T_1)$$

3) 查询满足要求的T₂中學生的姓名

$$\pi_{sn} (S \bowtie T_2)$$

➤ 将上述依次代入后得到最终的查询表达式:

$$\pi_{sn} (S \bowtie (\pi_{sno} (SC \bowtie (\pi_{cno} (\sigma_{sno=S5} (SC)))))))$$

学生 S (sno, sn, sd, sa)

课程 C (cno, cn, pno)

选课 SC (sno, cno, g)

【例5】检索修读S4所修读的所有课程的学生姓名

【分析】

1) 结果元组需要满足的条件？

修读了S4所修读过的所有课程

2) S4修读了哪些课程？

3) 如何根据一组课程 (S4所修读过的课程) 查出修读了其中的所有课程的学生元组？

【答】

$$\pi_{sn} (S \bowtie (\pi_{sno,cno}(SC) \div \pi_{cno} (\sigma_{sno=S4}(SC)))))$$

【例5】

$$\pi_{sn} (S \bowtie (\pi_{sno,cno}(SC) \div \pi_{cno} (\sigma_{sno=S4}(SC))))$$

【思考】 本题的以下几种表示方法都是错误的，为什么？

① $\pi_{sn} (S \bowtie \pi_{sno} (SC \bowtie \pi_{cno} (\sigma_{sno=S4} (SC))))$

② $\pi_{sn} (S \bowtie (SC \div \pi_{cno} (\sigma_{sno=S4} (SC))))$

③ $\pi_{sn} (S \bowtie (\pi_{sno,cno} (SC) \div \sigma_{sno=S4} (SC)))$

④ $\pi_{sn} (S \bowtie (SC \div \sigma_{sno=S4}(SC)))$

⑤ $\pi_{sn} ((S \bowtie SC) \div \pi_{cno} (\sigma_{sno=S4} (SC)))$

⑥ $\pi_{sn,cno} (S \bowtie SC) \div \pi_{cno} (\sigma_{sno=S4} (SC))$

学生 S (sno, sn, sd, sa)

课程 C (cno, cn, pno)

选课 SC (sno, cno, g)

请仔细理解例4和例5的查询语义及其相互之间的区别。

【例4】检索至少修读为S₅所修读的一门课的学生姓名

$$\pi_{sn} (S \bowtie \pi_{sno} (SC \bowtie \pi_{cno} (\sigma_{sno = S5} (SC)))))$$

【例5】检索修读S₄所修读的所有课程的学生姓名

$$\pi_{sn} (S \bowtie (\pi_{sno, cno} (SC) \div \pi_{cno} (\sigma_{sno = S4} (SC)))))$$

【思考】请用关系代数来表示下述查询。

- ① 检索所有计算机系学生都选修过的课程的课程号和课程名
- ② 检索所有选修《数据库》的学生都选修过的课程的编号

学生 S (sno, sn, sd, sa)

课程 C (cno, cn, pno)

选课 SC (sno, cno, g)

【例6】检索修读所有课程的学生学号

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

【思考】检索所有学生都选修过的课程的编号和名称

【例7】检索不修读任何课程的学生学号

$$\pi_{\text{sno}} (\text{S}) - \pi_{\text{sno}} (\text{SC})$$

插入、删除、修改操作的表示 1

【例8】在关系C中增添一门新课程

$$C \cup \{ (C_{13}, ML, C_3) \}$$

【例9】学号为 S_{17} 的学生因故退学,请在S及SC中将其除名

$$SC - (\sigma_{sno = s17} (SC))$$

$$S - (\sigma_{sno = s17} (S))$$

插入、删除、修改操作的表示 2

【例10】将关系S中学生 S_6 的年龄改为22岁

【答】

① 查询学号为 S_6 的学生: $T_1 := \sigma_{sno = s6}(S)$

② 修改后的学生 S_6 元组构成关系 W

③ 例3.23操作后的结果关系为 $(S - T_1) \cup W$

合并代入后如下:

$$(S - \sigma_{sno = s6}(S)) \cup W$$

【例11】将关系S中的年龄均增加1岁

$$S(sno, sn, sd, sa + 1)$$

复杂查询的表示

❑ 复杂查询

- 差运算的应用
- 笛卡尔积、 θ -连接、自然连接的应用
- 除运算的应用

❑ MyCAP数据库

- 顾客 (编号, 姓名, 城市, 折扣)
C(cid, cn, city, dis)
- 供应商 (编号, 名称, 城市, 佣金比例)
A(aid, an, city, per)
- 商品 (编号, 名称, 城市, 库存量, 单价)
P(pid, pn, city, qua, pri)
- 订单 (订单编号, 订购日期, 顾客编号, 供应商编号,
商品编号, 订购数量, 订单金额)
O(ordno, orddate, cid, aid, pid, qty, dol)

关系代数的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例1】检索没有经销过'p02'号商品的供应商的编号(aid)

$$\pi_{aid} (A) - \pi_{aid} (\sigma_{pid = 'p02'} (O))$$

关系代数中的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例2】检索只销售过‘p02’号商品的供应商的编号

$$\pi_{aid} (O) - \pi_{aid} (\sigma_{pid \neq 'p02'} (O))$$

? $\pi_{aid} (A) - \pi_{aid} (\sigma_{pid \neq 'p02'} (O))$ ✗

(注: \neq 是不等比较, 在关系代数中也可以使用 \neq 来表示不等比较)

关系代数中的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例3】 检索销售过 ‘c004’号顾客购买过的所有商品的供应商的编号 (aid)

$$\pi_{aid, pid} (O) \div \pi_{pid} (\sigma_{cid = 'c004'} (O))$$

【例4】 检索购买过 'p01' 和 'p07' 号两种商品的顾客的编号

$$\pi_{\text{cid}} (\sigma_{\text{pid} = \text{'p01'}} (O)) \cap \pi_{\text{cid}} (\sigma_{\text{pid} = \text{'p07'}} (O))$$

□ 请检查下述关系代数表达式，是否也能满足本题的查询要求？

[1] $\pi_{\text{cid}} (\sigma_{\text{pid} = \text{'p01'}} \text{ and } \text{pid} = \text{'p07'}} (O))$ ✗

[2] $\pi_{\text{cid}} (\sigma_{\text{pid} = \text{'p01'}} \text{ or } \text{pid} = \text{'p07'}} (O))$ ✗

[3] $\pi_{\text{cid}} (O \bowtie \pi_{\text{pid}} (\sigma_{\text{pid} = \text{'p01'}} \text{ or } \text{pid} = \text{'p07'}} (P)))$ ✗

[4] $\pi_{\text{cid}, \text{pid}} (O) \div \pi_{\text{pid}} (\sigma_{\text{pid} = \text{'p01'}} \text{ or } \text{pid} = \text{'p07'}} (P))$ ✓

[5] 令 $S := O$

$$\pi_{S.\text{cid}} ((\sigma_{\text{pid} = \text{'p01'}} (S)) \underset{S.\text{cid} = O.\text{cid}}{\bowtie} (\sigma_{\text{pid} = \text{'p07'}} (O)))$$
 ✓

[6] $\pi_{\text{cid}} (\sigma_{\text{pid} = \text{'p01'}} (O) \cap \sigma_{\text{pid} = \text{'p07'}} (O))$ ✗

【例5】检索购买过‘p01’或‘p07’号商品的顾客的编号

$$\pi_{cid} (\sigma_{pid = 'p01'} (O)) \cup \pi_{cid} (\sigma_{pid = 'p07'} (O))$$

□ 请检查下述关系代数表达式，是否也能满足本题的查询要求？

[1] $\pi_{cid} (\sigma_{pid = 'p01' \text{ and } pid = 'p07'} (O))$ ✗

[2] $\pi_{cid} (\sigma_{pid = 'p01' \text{ or } pid = 'p07'} (O))$ ✓

[3] $\pi_{cid} (O \bowtie \pi_{pid} (\sigma_{pid = 'p01' \text{ or } pid = 'p07'} (P)))$ ✓

[4] $\pi_{cid, pid} (O) \div \pi_{pid} (\sigma_{pid = 'p01' \text{ or } pid = 'p07'} (P))$ ✗

[5] $\pi_{cid} (\sigma_{pid = 'p01'} (O) \cup \sigma_{pid = 'p07'} (O))$ ✓

思考

查询满足下述条件的顾客的编号：

- ① 购买过 'p01' 号商品
- ② 没有购买过 'p01' 号商品
- ③ 只购买过 'p01' 号商品
- ④ 既购买过 'p01' 也购买过 'p02' 号商品
- ⑤ 既没有购买过 'p01'，也没有购买过 'p02' 号商品
- ⑥ 只购买过 'p01' 和 'p02' 号两种商品
- ⑦ 没有购买过商品
- ⑧ 只购买过一次商品
- ⑨ 只购买过同一种商品

关系代数的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例6】 检索购买过 'p02' 号商品的顾客所在的城市
及 经销过 'p02' 号商品的供应商所在城市

$$\pi_{\text{city}}((\sigma_{\text{pid}='p02'}(O)) \bowtie C) \cup \pi_{\text{city}}((\sigma_{\text{pid}='p02'}(O)) \bowtie A)$$

关系代数中的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例7】 检索至少为购买过‘p01’号商品的某一位顾客
订购过商品 的 供应商的编号 (aid)

$$\pi_{aid} \left(\left(\pi_{cid} \left(\sigma_{pid='p01'} (O) \right) \right) \bowtie O \right)$$

关系代数的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例8】 检索为购买过 ‘p01’ 号商品的所有顾客
订购过商品的供应商的编号 (aid)

$$\pi_{cid, aid} (O) \div \pi_{cid} (\sigma_{pid = 'p01'} (O))$$

关系代数中的应用示例

顾客 $C(cid, cn, city, dis)$

供应商 $A(aid, an, city, per)$

商品 $P(pid, pn, city, qua, pri)$

订单 $O(ordno, orddate, cid, aid, pid, qty, dol)$

【例9】 检索没有被居住在以字母‘D’打头的城市中的顾客购买过的商品的编号(pid)

$$1) \quad T_1 := \sigma_{city \geq 'D' \text{ and } city < 'E'} (C)$$

$$2) \quad T_2 := \pi_{pid} (P) - \pi_{pid} (O \bowtie T_1)$$

关系代数中的应用示例

顾客 C (cid, cn, city, discnt)

【例10】 检索具有最高折扣(discent)的顾客的编号(cid)

【解1】 令 $M := C, N := C$

$$1) T_1 := \pi_{N.cid} (M \underset{M.discnt > N.discnt}{\bowtie} N)$$

$$2) T_2 := \pi_{cid} (C) - T_1$$

【解2】 令 $M := C, N := C$

$$1) T_1(cyid, cid) := \pi_{M.cid, N.cid} (M \underset{M.discnt \geq N.discnt}{\bowtie} N)$$

$$2) T_2 := T_1 \div \pi_{cid} (C)$$

【例10】检索具有最高折扣(discont)的顾客的编号(cid)

□ 也可不使用 θ -联接运算，直接使用笛卡尔积实现两个关系合并

【解3】 令 $M := C, N := C$

$$1) T_1 := \pi_{N.cid} (\sigma_{M.discont > N.discont} (M \times N))$$

$$2) T_2 := \pi_{cid} (C) - T_1$$

【解4】 令 $M := C, N := C$

$$1) T_1 (cyid, cid) := \pi_{M.cid, N.cid} (\sigma_{M.discont \geq N.discont} (M \times N))$$

$$2) T_2 := T_1 \div \pi_{cid} (C)$$

关系代数的应用示例

顾客 **C**(cid, cn, city, dis)

供应商 **A**(aid, an, city, per)

商品 **P**(pid, pn, city, qua, pri)

订单 **O**(ordno, orddate, cid, aid, pid, qty, dol)

【例11】 检索 至少购买过一种价格为\$0.50的商品的
顾客的名称 (cn)

$$\pi_{cn} \left(\left(\left(\pi_{pid} \left(\sigma_{pri=0.5} (P) \right) \right) \bowtie O \right) \bowtie C \right)$$

关系代数中的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例12】 检索至今为止没有通过‘a03’号供应商来购买过商品的顾客的编号 (cid)

$$\pi_{cid} (C) - \pi_{cid} (\sigma_{aid = 'a03'} (O))$$

$$? \pi_{cid} (O) - \pi_{cid} (\sigma_{aid = 'a03'} (O)) \quad \times$$

关系代数中的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例13】 检索仅通过 ‘a03’号供应商来购买商品
的顾客的编号 (cid)

$$\pi_{cid} (O) - \pi_{cid} (\sigma_{aid \neq 'a03'} (O))$$

$$? \pi_{cid} (C) - \pi_{cid} (\sigma_{aid \neq 'a03'} (O)) \quad \times$$

关系代数的应用示例

顾客 **C**(cid, cn, city, dis)

供应商 **A**(aid, an, city, per)

商品 **P**(pid, pn, city, qua, pri)

订单 **O**(ordno, orddate, cid, aid, pid, qty, dol)

【例14】 检索从未被居住在 ‘New York’ 的顾客通过位于 ‘Boston’ 的供应商购买过的商品的编号 (pid)

$$1) \quad T_1 := \pi_{cid} (\sigma_{city='New York'} (C))$$

$$2) \quad T_2 := \pi_{aid} (\sigma_{city='Boston'} (A))$$

$$3) \quad T_3 := \pi_{pid} ((T_1 \bowtie O) \bowtie T_2)$$

$$4) \quad T_4 := \pi_{pid} (P) - T_3$$

关系代数的应用示例

顾客 **C**(cid, cn, city, dis)

供应商 **A**(aid, an, city, per)

商品 **P**(pid, pn, city, qua, pri)

订单 **O**(ordno, orddate, cid, aid, pid, qty, dol)

【例15】 检索购买过所有价格为\$0.50的商品
的顾客的姓名 (cn)

$$1) \quad T_1 := \pi_{cid,pid} (O) \div \pi_{pid} (\sigma_{pri=0.50} (P))$$

$$2) \quad T_2 := \pi_{cn} (T_1 \bowtie C)$$

关系代数的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例16】 检索购买过所有被销售过的商品的顾客的编号 (cid)

$$\pi_{cid,pid} (O) \div \pi_{pid} (O)$$

关系代数的应用示例

顾客 C(cid, cn, city, dis)

供应商 A(aid, an, city, per)

商品 P(pid, pn, city, qua, pri)

订单 O(ordno, orddate, cid, aid, pid, qty, dol)

【例17】 检索至少通过经销过'p03'号商品的供应商之一去购买过商品的顾客的编号 (cid)

$$\pi_{cid} ((\pi_{aid} (\sigma_{pid='p03'} (O))) \bowtie O)$$

关系代数的应用示例

顾客 $C(cid, cn, city, dis)$

供应商 $A(aid, an, city, per)$

商品 $P(pid, pn, city, qua, pri)$

订单 $O(ordno, orddate, cid, aid, pid, qty, dol)$

【例18】 检索与居住在 ‘Dallas’ 或 ‘Boston’ 的某一位顾客具有相同折扣的顾客的编号 (cid)

$$\pi_{cid} \left(\left(\pi_{dis} \left(\sigma_{city='Dallas' \text{ or } city='Boston'} (C) \right) \right) \bowtie C \right)$$

关系代数的应用

- ❑ MyCAP数据库
- ❑ 基本运算符的查询表示
- ❑ 扩充运算符的查询表示
- ❑ 小结

关系代数总结

□ 关系的表示

- 元组的集合

□ 关系操纵的表示

- 集合上的运算

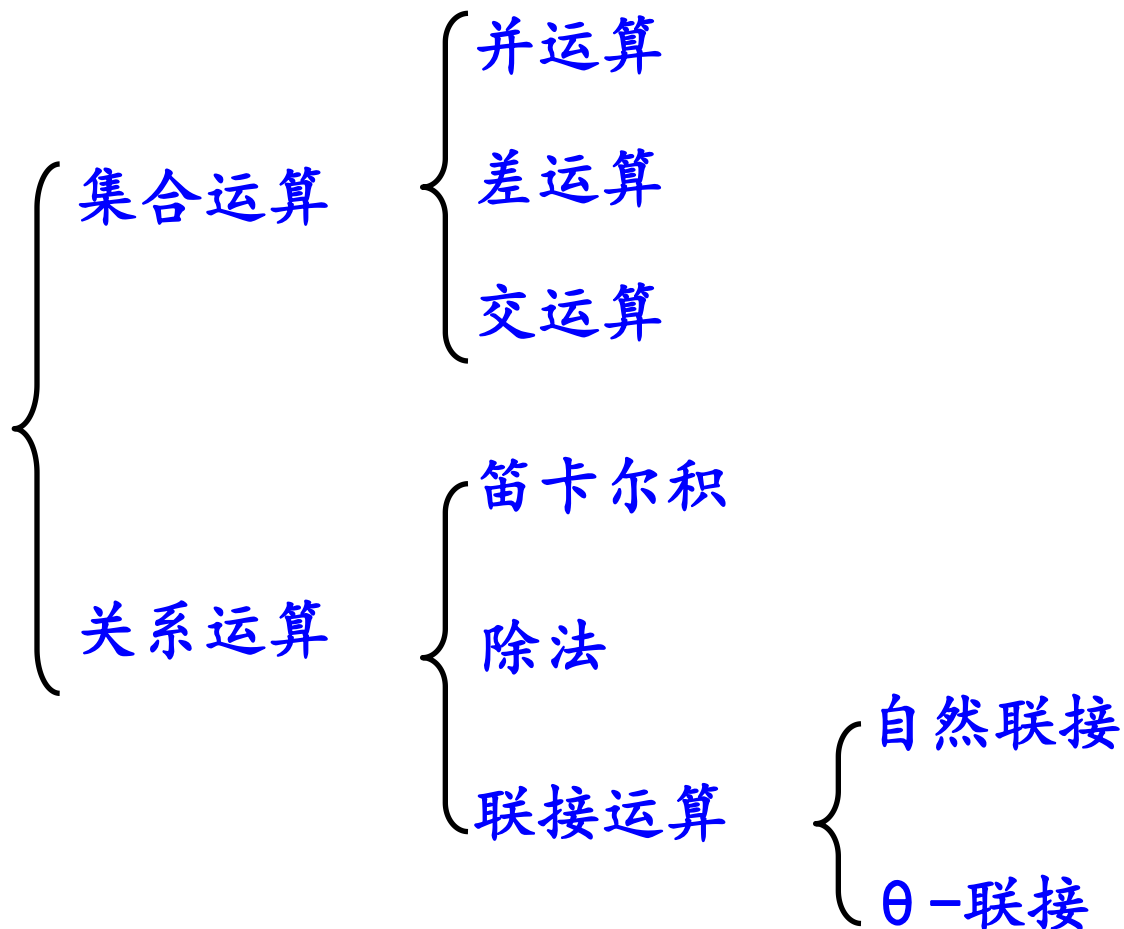
□ 五种基本运算

- 并 \cup ，差 $-$
- 投影 π ，选择 σ ，笛卡儿积 \times

□ 五种扩充运算

- 交 \cap
- 除 \div
- θ -联接，自然联接 \bowtie ，外联接

关系的合并运算



关系代数运算符的使用

□ When & How

- 投影 / 选择
- 并 / 交 / 差
- 笛卡尔积 / θ -联接 / 自然联接 / 除法

□ Difference

- 笛卡尔积 vs. 自然联接
- 自然联接 vs. 除法
- 差运算 vs. ‘不等’比较 (not equal)

特殊运算（差）的使用方法

□ When ?

- 当查询条件带有‘否定’语义，或者具有明显的‘排它性’的时候，通常需要使用两个子查询之间的‘差’运算

□ How ?

- ‘差’运算的运算对象（关系）中，通常需要包含其关键字

“笛卡尔积/ θ -联接/自然联接”的使用方法

□ 都是关系的合并运算

- 笛卡尔积是基本运算， θ -联接和自然联接则是扩充运算
- 请注意三者的结果关系的关系模式之间的区别

□ 笛卡尔积

- 是实现跨不同关系表进行数据访问的基础
- 在笛卡尔积的结果关系中，存在着很多无意义的结果元组，通常需要通过后续的选择运算过滤掉

□ θ -联接

- 相邻的“笛卡尔积+选择运算”可以合并为一个 θ -联接

□ 自然联接

- 如果联接条件是基于“两张表中的所有同名属性的相等比较”，可以将 θ -联接进一步简写为自然联接

“笛卡尔积/ θ -联接/自然联接” (cont.)

❑ 一般方法：笛卡尔积+选择 or θ -联接

- 不存在同名属性，或者联接条件不是基于同名属性的相等比较
- 在结果关系中可能存在同名属性，需要加以区别

❑ 常用方法：自然联接

- 联接条件是隐含的（所有同名属性的相等比较）
- 如果在两个关系之间存在多对‘同名属性’，而本次查询又不需要‘所有’的同名属性都相等，此时有两种选择：
 - ① 采用前述的一般方法来实现关系的合并
 - ② 先对其中的一个关系执行投影运算，过滤掉其中不需要相等的那些同名属性，然后再使用自然联接运算

❑ 难点：关系的自联接

- 使用赋值运算定义‘同质不同名’的两个中间关系（元组集合相同，但关系名不同），当然也可以对中间关系中的属性进行重命名
- 然后再使用前述的一般方法实现两个中间关系的合并

特殊运算（除）的使用方法

□ ‘除’ 运算与 ‘联接’ 运算的区别

- 我们将查询的结果关系称为‘目标对象’，用于定义查询条件的关系称为‘条件对象’
- 在决定某个元组t是否属于结果关系时，
 - 如果只需要从条件对象中找到一个元组c并使得查询条件成立，那么就直接使用‘联接’运算（包括笛卡尔积、 θ -联接和自然联接）
 - 如果需要条件对象集中的所有元组都能使得查询条件成立，那么就使用‘除’运算

□ ‘除’ 运算表达式的表示方法

- 被除数关系中必须包含目标对象和条件对象的关键字
- 除数关系中只含条件对象的关键字
- 被除数和除数关系中不能含其它‘不必要’的多余属性

复习思考题 1

1. 请用五种基本运算分别写出下列两个扩充运算的推导公式：
① ‘自然联接’运算 ② ‘除’运算
2. 理解笛卡尔积、 θ -联接、自然联接三者之间的联系和区别，并掌握：
① 在何种场景下适合使用自然联接？
② 如何正确使用‘自然联接’运算。
3. 如何理解‘除’运算的语义？请解释其推导公式的含义，并掌握如何正确书写‘除’运算表达式。
4. 理解 笛卡儿积/ θ -联接/自然联接 和 ‘除’运算之间的区别，并回答：
① 在什么情况下适合应用‘除’运算而不是联接运算？请举一例并写出使用除法和不使用除法的两种查询表达式。
② 请列举一个与情况①相似但又不能使用除法的例子，并写出其查询表达式。
5. 什么是关系的自联接？请列举一个使用自联接的例子，并写出查询表达式。
6. 什么是外联接？请列举一个使用外联接的例子，并写出查询表达式。

复习思考题 2

7. 设有一个公司产品零售数据库，其关系模式如下(带下划线的属性是关键字;以订单编号ordno的大小来区分订单的先后, ordno小的在前)。请用关系代数分别写出下述查询。

关系名	属性集	关系模式
顾客	顾客编号, 姓名, 居住城市, 折扣	C (<u>cid</u> , cname, city, discnt)
供应商	供应商编号, 名称, 所在城市, 佣金比例	A (<u>aid</u> , aname, city, percent)
商品	商品编号, 名称, 库存城市, 库存数量, 单价	P (<u>pid</u> , pname, city, stqty, price)
订单	订单编号, 订购日期, 顾客编号, 供应商编号, 商品编号, 订购数量, 销售金额	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

(1) 查询满足条件的顾客的编号cid:

- ① 只购买过 p01 和 p02 两种商品;
- ② “只购买过商品p01” 或者 “只购买过商品p02”;
- ③ 只购买过一种商品。

(2) 查询满足条件的供应商的名字aname: 没有销售过商品;

(3) 查询满足条件的供应商的编号aid:

- ① 只向供应商自己所在城市中的顾客销售过商品;
- ② 在所有有顾客的城市中都向客户销售过商品;
- ③ 向供应商自己所在城市中的所有顾客都销售过商品。

复习思考题 3

关系名	属性集	关系模式
顾客	顾客编号, 姓名, 居住城市, 折扣	C (<u>cid</u> , cname, city, discent)
供应商	供应商编号, 名称, 所在城市, 佣金比例	A (<u>aid</u> , aname, city, percent)
商品	商品编号, 名称, 库存城市, 库存数量, 单价	P (<u>pid</u> , pname, city, stqty, price)
订单	订单编号, 订购日期, 顾客编号, 供应商编号, 商品编号, 订购数量, 销售金额	O (<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

- (4) 在每一个供应商自己的所有订单中，查询销售金额dols最高的订单，结果返回供应商编号、该供应商的销售金额最高订单的订单编号和销售金额。
- (5) 查询每一个顾客的第一份订单和最后一份订单（在第①和第②小题中，不考虑没有订单和只有一份订单的顾客；在第③小题中，需要返回所有顾客的查询结果）
- ① 结果关系中只有 cid, ordno, orddate 三个属性；
 - ② 结果关系中有顾客编号cid, 第一份订单的ordno和orddate, 最后一份订单的ordno和orddate等五个属性；
 - ③ 结果关系的关系模式同上面的第②小题，每一个顾客返回一条查询结果。如果是没有订单的顾客，结果元组上只有该顾客的编号；如果是只有一份订单的顾客，那么这份订单就作为该顾客的最后一份订单，他的第一份订单信息为空。（提示：使用外联接）

复习思考题 4

关系	属性集	关系模式
学生	<u>学号</u> , 学生姓名, 就读院系, 年级	S (<u>sno</u> , sname, dept, grade)
课程	<u>课程号</u> , 课程名, 开课院系, 课程类型	C (<u>cno</u> , cname, dept, opt)
教师	<u>教师工号</u> , 教师姓名, 工作院系	T (<u>tno</u> , tname, dept)
选课	<u>学号</u> , <u>课程号</u> , 授课教师工号, 成绩, 修读年份	L (<u>sno</u> , <u>cno</u> , tno, score, years)

8. 设有一个如上所示的学生选课数据库，其中：带下划线的属性是关键字；课程类型分为‘必修’和‘选修’；同一门课同一个学生只能有一条选课记录；成绩采用百分制。请用关系代数写出下述查询。

查询满足下述条件的学生的学号和姓名：

- ① 只修读过自己就读院系开设的课程；
- ② 所有的修读课程成绩都合格（成绩 ≥ 60 ）；
- ③ 修读过‘计算机’系开设的所有‘必修’课程；
- ④ 修读了自己就读院系开设的所有‘必修’课程且在这些必修课程上的成绩都合格；
- ⑤ 2019级并且还没有修读通过自己就读院系开设的所有‘必修’课程。
（存在自己就读院系开设的‘必修’课程，该同学还没有修读，或者虽然修读但成绩没达到合格。）