

CHAP 2

1. (a) $\Pi_{SNO}(\sigma_{JNO='J1'}(SPJ))$

(b) $\Pi_{SNO}(\sigma_{JNO='J1' \wedge PNO='P1'}(SPJ))$

(c) $\Pi_{SNO}(\sigma_{JNO='J1' \wedge COLOR='red'}(SPJ \bowtie P))$
或 $\Pi_{SNO}(\Pi_{SNO,PNO}(\sigma_{JNO='J1'}(SPJ)) \bowtie \Pi_{PNO}(\sigma_{COLOR='red'}(P)))$

(d) $\Pi_{JNO}(\sigma_{\neg(CITY='天津' \wedge COLOR='红')}(SPJ \bowtie S \bowtie P))$
1) 用左外连接, 获得 SPJ 所有元组的供应商所在城市和零件颜色
2) 对所有元组进行筛选
3) 对 2 结果进行投影

$$\Pi_{JNO}(J) - \Pi_{JNO}(\Pi_{SNO}(\sigma_{city='天津'}(S)) \bowtie \Pi_{SNO,PNO,JNO}(SPJ) \bowtie \Pi_{PNO}(\sigma_{COLOR='red'}(P)))$$

- 1) 找出天津的供应商
- 2) 和 SPJ 表内连接, 获得天津供应商的零件号和项目号
- 3) 筛选出红色的零件和 2 进行内连接运算, 得到天津供应商的红色零件的项目号
- 4) 对 3 进行投影, 找出 3 的 JNO
- 5) 用集合差找出非 4 的元组

(e) 这题有歧义:
“至少使用了 S1 提供的全部零件的项目” 或 “至少使用了 S1 提供的全部零件型号的项目”
解题思路是:

- 1) 先获得 SPJ 中所有工程及其使用的零件, 得关系 R
- 2) 再获得所有由 S1 提供零件的 PNO (选择 + 投影), 得关系 S
- 3) $R \div S$, 所得是使用了 S1 全部零件的项目 JNO

$$\Pi_{JNO,PNO,SNO}(SPJ) \div \Pi_{SNO,PNO}(\sigma_{SNO='s1'}(SPJ))$$

$$\Pi_{JNO,PNO}(SPJ) \div \Pi_{PNO}(\sigma_{SNO='s1'}(SPJ))$$

$$\Pi_{JNO,PNO}(\sigma_{SNO='s1'}(SPJ)) \div \Pi_{PNO}(\sigma_{SNO='s1'}(SPJ))$$

```
mysql> select * from s;
```

A	B	C	D
5	4	3	2
9	9	7	8
8	9	1	6
8	9	9	9

4 rows in set (0.00 sec)

(a)

```
mysql> SELECT * FROM T;
```

C	D	E	F
3	2	5	4
7	3	2	4
7	8	7	8
3	3	9	9

4 rows in set (0.00 sec)

(b)

Figure 1: 本题用表格

2. (a) SELECT * FROM S WHERE A=10

(b) SELECT DISTINCT A, B FROM S;
说明: DISTINCT 关键字如果不加, 不扣分

(c) SELECT * FROM S NATURAL JOIN T;
说明: 自然连接, 它要求两个关系中进行比较的分量必须是同名的属性组, 并且在结束中把重复的属性列去掉。SQL 的自然连接, 可参看战老师 708 视频。自然连接无需指定连接列, SQL 会检查两个表中是否相同名称的列, 且假设他们在连接条件中使用。自然连接不允许使用 ON 语句 (注意 Using 的用法)

```
mysql> select * from s natural join t;
```

C	D	A	B	E	F
3	2	5	4	5	4
7	8	9	9	7	8

2 rows in set (0.00 sec)

Figure 2: 自然连接中, 重复列被删除.

也可以是:

SELECT A,B,S.C,S.D,E,F FROM S,T WHERE S.C=T.C AND S.D=T.D;
结果一样, 注意输出字段的排列顺序, 此时被指定

(d) SELECT * FROM S inner join T on S.C=T.C; 或
SELECT * FROM S , T
WHERE S.C=T.C;
说明: 该连接为 Θ 连接中的等值连接

```
mysql> SELECT * FROM S inner join T on S.C=T.C;
```

A	B	C	D	C	D	E	F
5	4	3	2	3	2	5	4
9	9	7	8	7	3	2	4
9	9	7	8	7	8	7	8
5	4	3	2	3	3	9	9

4 rows in set (0.00 sec)

Figure 3: 等值连接中重复列不删除.

- (e) `SELECT * FROM S , T`
`WHERE S.A<T.E;`

说明: 该连接为 Θ 连接中的不等值连接, 先做笛卡尔积, 再选择满足条件的记录

```
mysql> SELECT * FROM S , T
-> WHERE S.A<T.E;
```

A	B	C	D	C	D	E	F
5	4	3	2	7	8	7	8
5	4	3	2	3	3	9	9
8	9	1	6	3	3	9	9
8	9	9	9	3	3	9	9

Figure 4: 不等值连接.

- (f) `SELECT * FROM (SELECT C,D FROM S) AS K, T;`

说明: 该要求为先做 S 的投影, 将结果再与 T 做笛卡尔积。由于涉及到先后次序, 可以考虑用嵌套查询来实现。在最新的 SQL 标准中, 嵌套语句可以出现在任意位置

```
mysql> SELECT * FROM (SELECT C,D FROM S) AS K, T;
```

C	D	C	D	E	F
3	2	3	2	5	4
7	8	3	2	5	4
1	6	3	2	5	4
9	9	3	2	5	4
3	2	7	3	2	4
7	8	7	3	2	4
1	6	7	3	2	4
9	9	7	3	2	4
3	2	7	8	7	8
7	8	7	8	7	8
1	6	7	8	7	8
9	9	7	8	7	8
3	2	3	3	9	9
7	8	3	3	9	9
1	6	3	3	9	9
9	9	3	3	9	9

16 rows in set (0.00 sec)

Figure 5: 先投影后笛卡尔积.

3. (a) 找出所有供应商的姓名和所在城市
 SELECT SNAME,CITY
 FROM S;

```
mysql> SELECT SNAME,CITY
-> FROM S;
```

SNAME	CITY
精益	天津
盛锡	北京
东方红	北京
丰盛泰	天津
为民	上海

```
5 rows in set (0.00 sec)
```

Figure 6: 投影.

- (b) 找出所有零件的名称颜色和重量
 说明：所有的信息都在 P 表，所以是单表查询。而且找出所有，所以是投影。
 SELECT PNAME,COLOR,WEIGHT
 FROM P;

```
mysql> SELECT PNAME,COLOR,WEIGHT
-> FROM P;
```

PNAME	COLOR	WEIGHT
螺母	红	12
螺栓	绿	17
螺丝刀	蓝	14
螺丝刀	红	14
凸轮	蓝	40
齿轮	红	30

```
6 rows in set (0.00 sec)
```

Figure 7: 投影.

- (c) 找出使用供应商 S1 所供应零件的工程号码
 说明：所有的信息都在 SPJ 表，所以是选择 + 投影。
 SELECT DISTINCT JNO
 FROM SPJ
 WHERE SNO='S1';

```
mysql> SELECT DISTINCT JNO
-> FROM SPJ
-> WHERE SNO='S1';
```

JNO
J1
J3
J4
J2

Figure 8: 单表选择 + 投影，注意 DISTINCT.

- (d) 找出工程 J2 使用各种零件的名称及其数量
 说明：工程 J2 的信息在 SPJ 表，零件信息在 P 表，所以是选择 + 连接 + 投影。
 SELECT PNAME,QTY
 FROM SPJ INNER JOIN P ON SPJ.PNO=P.PNO
 WHERE JNO='J2';

```
mysql> SELECT PNAME,QTY
-> FROM SPJ INNER JOIN P ON SPJ.PNO=P.PNO
-> WHERE JNO='J2';
```

PNAME	QTY
螺栓	100
螺丝刀	200
凸轮	100
齿轮	100

Figure 9: INNER JOIN

- (e) 找出上海厂商供应的所有零件号码
 说明：工程信息在 SPJ 表，厂商信息在 S 表，所以是选择 + 连接 + 投影。**DITINCT 不能缩写，不能省略**
 SELECT DISTINCT PNO
 FROM S INNER JOIN SPJ ON S.SNO=SPJ.SNO
 WHERE S.CITY='上海';

```
mysql> SELECT DISTINCT PNO
-> FROM S INNER JOIN SPJ ON S.SNO=SPJ.SNO
-> WHERE S.CITY='上海';
```

PNO
P2
P3
P6

Figure 10: 注意 DISTINCT

- (f) 找出使用上海产的零件的工程名称

说明：工程信息在 SPJ 表，厂商信息在 S 表，工程名称在 J 表。所以是选择 + 连接 + 投影。

```
SELECT JNAME
FROM J
WHERE JNO IN (SELECT DISTINCT JNO
FROM S INNER JOIN SPJ ON S.SNO=SPJ.SNO
WHERE S.CITY='上海');
```

```
mysql> SELECT JNAME
-> FROM J
-> WHERE JNO IN (SELECT DISTINCT JNO
-> FROM S INNER JOIN SPJ ON S.SNO=SPJ.SNO
-> WHERE S.CITY='上海');
```

JNAME
造船厂
三建
二汽

Figure 11: 三表操作

- (g) 找出没有使用天津产的工程零件的工程号码

说明：可以考虑 NOT EXISTS 关键字。本例中结果为空集

```
SELECT JNO
FROM SPJ
WHERE NOT EXISTS
(SELECT *
FROM SPJ AS B INNER JOIN S ON B.SNO=S.SNO
WHERE SPJ.JNO= B.JNO AND S.CITY='天津');
```

```
mysql> SELECT JNO
-> FROM SPJ
-> WHERE NOT EXISTS (SELECT *
-> FROM SPJ INNER JOIN S ON SPJ.SNO=S.SNO
-> WHERE S.CITY='天津');
Empty set (0.00 sec)
```

Figure 12: NOT EXISTS 关键字

```
SELECT JNO
FROM J /* 从 J 表入手，会包含尚未使用任何零件的工程号 */
WHERE NOT EXISTS
(SELECT *
FROM SPJ ,S
WHERE SPJ.JNO= J.JNO
AND SPJ.SNO=S.SNO
AND S.CITY='天津');
```

- (h) 把所有红色零件的颜色改成蓝色

说明：UPDATE 语句

UPDATE P SET COLOR=' 蓝' WHERE COLOR=' 红';

```
mysql> UPDATE P
-> SET COLOR=' 蓝'
-> WHERE COLOR=' 红';
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

Figure 13: 观察结果有几条记录被更新

- (i) 由 S5 供给 J4 的零件 P6 改为由 S3 供应

说明：注意要把所有条件用上

UPDATE SPJ

SET SNO='S3'

WHERE JNO='J4' AND SNO='S5' AND PNO='P6';

- (j) 从供应商关系中删除 S2 的记录，并从供应情况关系中删除相应的记录

说明：注意删除顺序

DELETE FROM SPJ

WHERE SNO='S2';

DELETE FROM S

WHERE SNO = 'S2';

- (k) 请将 (S2,J6,P4,200) 插入 SPJ

说明：注意没有指定插入位置，另外属性列一定要人为将其于表属性顺序一致，且插入的值与属性顺序一一对应，必要时加引号。

INSERT INTO SPJ

VALUES(S2,P4,J6,200);