# Chapter 2

## 2.6

RESULTS

| | sno |
|---|---|
| 1 | S1 |
| 2 | S2 |
| 3 | S2 |
| 4 | S3 |
| 5 | S3 |
| 6 | S4 |
| 7 | S5 |

| | sno |
|---|---|
| 1 | S1 |
| 2 | S3 |

| | sno |
|---|---|
| 1 | S1 |
| 2 | S3 |

| | jno |
|---|---|
| 1 | J2 |
| 2 | J5 |
| 3 | J6 |
| 4 | J7 |

| | jno |
|---|---|
| 1 | J1 |
| 2 | J4 |

2

## 2.7 , ## 2.8 $\sigma, \times, \Pi, -$ ( , , )

$$A \cap B = A - (A - B)$$

$$\neg A = U - A$$

$$A \cup B = U - (\neg A) \cap (\neg B)$$

$$A \bowtie_{cond} B = \sigma_{cond}(A \times B)$$

$$A \bowtie B = \Pi_{attr(A) \cup attr(B)}(\sigma_{A[attr(A) \cap attr(B)] = B[attr(A) \cap attr(B)]}(A \times B))$$

# Chapter 3 ## 3.3

```
drop table if exists S;
drop table if exists T;

create table S(
  A int,
  B int,
  C int,
  D int
);

create table T(
  C int,
  D int,
  E int,
  F int
);

insert into S values
(1, 2, 3, 4),
(2, 2, 3, 4),
(3, 2, 4, 4),
(1, 2, 4, 4);

insert into T values
(3, 4, 3, 4),
(4, 4, 3, 4),
(3, 2, 4, 4),
(1, 2, 4, 4);

-- 1
select * from S where A=10;

-- 2
select A,B from S;

-- 3
```

```
select * from S inner join T on (S.C=T.C and S.D=T.D);

-- 4
select A, B, S.C as C1, S.D as D1, T.C as C2, T.D as D2, E, F
  from S inner join T on (S.C=T.C);

-- 5
select A, B, S.C as C1, S.D as D1, T.C as C2, T.D as D2, E, F
  from S, T where (A<E);

-- 6
select TMP.C as C1, TMP.D as D1, T.C as C2, T.D as D2, E, F
  from (select distinct C, D from S) as TMP, T;
```

### 3.4

```
/* 1 */
select sno
from SPJ
where jno='J1';
/* 2 */
select sno
from SPJ
where jno='J1' and pno='P1';
/* 3 */
select sno
from SPJ
where
  jno = 'J1' and pno in (select pno
  from P
  where color=' ');
/* 4 */
-- tianjiSupplier <- select sno from S where city=' ';
-- redPart <- select pno from P where color=' ';
select jno
from J
where not exists(
  select *
from SPJ
where
    sno in (select sno
  from S
  where city=' ') and pno in (select pno
  from P
  where color=' ') and J.jno = SPJ.jno
```

```
);
/* 5 */
select distinct SPJ.jno
from SPJ inner join (
    select distinct TMP.pno
  from SPJ as TMP
  where TMP.sno='S1'
  ) as REF on (SPJ.pno = REF.pno)
group by SPJ.jno
having count(SPJ.pno) in (
    select count(distinct TMP.pno)
from SPJ as TMP
where TMP.sno='S1'
  )
-- go
-- select * from SPJ inner join S1Part on (S1Part.pno=SPJ.pno);
```

### 3.5

“‘sql – 1 select sname, city from S;

– 2 select pname, color, weight from P;

– 3 select distinct jno from SPJ where sno=‘S1’;

– 4 select pname, sum(qty) as sum_qty from P, SPJ where P.pno = SPJ.pno and jno = ‘J2’ group by pname;

– 5 select distinct pno from S, SPJ where city=‘ ’ and S.sno = SPJ.sno;

– 6 select distinct jname from S, SPJ, J where S.sno = SPJ.sno and J.jno = SPJ.jno and S.city = ‘ ’;

– 7 select J.jno from J where not exists( select * from SPJ, S where J.jno = SPJ.jno and SPJ.sno = S.sno and S.city = ‘ ’ );

– 8 update P set color=‘ ’ where color=‘ ’;

– 9 update SPJ set sno=‘S3’ where sno=‘S5’ and jno=‘J4’ and pno=‘P6’;

– 10 /* on delete cascade */ delete from S where sno=‘S2’;

– 11 insert into SPJ value(‘S2’, ‘P4’, ‘J6’, 200);

`#Chapter 4 ## 4.6`sql – 4.6.1 grant all priviledges on table Student to U1; with grant option; – 4.6.2 grant select, update(addr) on table Student to U2; – 4.6.3 grant select on table Class to public; – 4.6.4 grant update, select on table Student to R1; – 4.6.5 grant R1 to U1 with grant option; “‘ ## 4.7

```
sql -- 4.7.1 grant select on table Staff, Department to WangMing;
-- 4.7.2 grant insert, delete on table Staff, Department to
```

```
LiYong; -- 4.7.3 ---- BUGGY grant select on Staff when User()=name
to public; go ---- Worked create view hole as   select *   from
Staff   where OWNER=user; go grant select on hole to public; --
4.7.4 grant select, update(Salary) on table Staff to LiuXing; --
4.7.5 grant references on table Staff, Department to ZhangXin;
-- 4.7.6 grant all priviledges on table Staff, Department to
ZhouPing with grant option; go -- 4.7.7 create view stat as
select depart_id, max(Salary), min(Salary), average(Salary)
from Staff    group by depart_id; go grant select on stat to
YangLan; go# Chapter 5 ## 5.6
```

```
--
create table Department(
  dpid char(20) primary key,
  dname char(20),
  manager char(20),
  phone char(20)
);

create table Staff(
  sid char(20) primary key,
  sname char(20),
  age char(20),
  title char(20),
  salary int,
  dpid char(20) REFERENCES Department(dpid),
  CONSTRAINT CK_AGE_LE_60 check(age <= 60)
);
```

## 5.7

, , constraint , , , .