- 3.1

  a.

```sql
select * from course where dept_name = 'Comp. Sci.' and credits = 3;
```

b.

```sql
select distinct c.id
from instructor natural join teaches join takes c using(year,semester,course_id,sec_id)
where name = 'Einstein';
```

c.

```sql
select max(salary) from instructor;
```

d.

```sql
select * from instructor where salary = (select max(salary) from instructor);
```

e.

```sql
select course_id,sec_id,count(*)
from takes
where year = 2017 and semester = 'Fall'
group by course_id,sec_id;
```

f.

```sql
select max(cnt)
from(
select count(*) cnt
from takes
where year = 2017 and semester = 'Fall'
group by course_id,sec_id);
```

g.

```sql
with ta as(select course_id,sec_id,count(*) cnt
from takes
where year = 2017 and semester = 'Fall'
group by course_id,sec_id)
```

```sql
select course_id,sec_id
from ta
where cnt = (select max(cnt) from ta);

```

- 3.2 a.

```sql
select sum(points*credits)
from takes natural join grade_points natural join course
where id = '12345';
```

b.

```sql
select sum(points*credits) / sum(credits) gpa
from takes natural join grade_points natural join course
where id = '12345';
```

c.

```sql
select id,sum(points*credits) / sum(credits) gpa
from takes natural join grade_points natural join course
group by id;
```

d.

```sql
select id,sum(points*credits) / sum(credits) gpa
from takes natural join grade_points natural join course
group by id;
```

e.

如果成绩为空，那么这些课程不会进入统计里面，如果需求如此，那么对上述查询没有什么影响。如果需要纳入统计的话，需要使用外联 grade_points 表，并将成绩为空视为 0, 进行处理。

比如 a 改为:

```sql
select sum(case when points is null then 0 else points end * credits)
from takes natural left outer join grade_points natural join course
where id = '12345';
```

其他做类似修改即可。

- 3.3 a.

```sql
update instructor set salary = salary * 1.1 where dept_name = 'Comp. Sci.';
```

b.

```sql
DELETE FROM course a
WHERE NOT EXISTS
(SELECT 1
FROM section b
WHERE a.course_id = b.course_id)
```

c.

```
insert into instructor
select id,name,dept_name,10000
from student
where tot_cred > 100;
```

- 3.4 a.

```
SELECT COUNT (*)
FROM person a
WHERE EXISTS
(SELECT 1
FROM owns b NATURAL JOIN participated c
WHERE a.driver_id = b.driver_id)
```

b.

```
delete from owns a
where a.driver_id = '12345'
and exists(select 1 from car b
where a.license_plate = b.license_plate
and b.year = 2010);
```

- 3.5

```
select id,score,
case when score < 40 then 'F'
when score < 60 then 'C'
when score < 80 then 'B'
else 'A' end grade
from marks;
```

- 3.6

```
lower(name) like '%sci%'
```

- 3.7

$p \subseteq r1 \cup r2 \ and \ r1 \cap r2 = \Phi$

- 3.8 a.

```
select distinct customer_name
from depositor a
```

```
where not exists(select 1 from borrower b
where a.customer_name = b.customer_name);
```

b.

```
select * from customer a
where exists(select 1 from customer  b
where a.customer_street = b.customer_street
and a.customer_city = b.customer_city
and b.id = '12345');
```

c.

```
select distinct branch_name
from account a natural join depositor b natural join customer c
where c.customer_city = 'Harrison';
```

- 3.9 a.

```
select id,person_name,customer_city
from employee a
where exists(select 1 from works b
where a.id = b.id and b.company_name = 'First Bank Corporation');
```

b.

```
select id,person_name,customer_city
from employee a
where exists(select 1 from works b
where a.id = b.id and b.company_name = 'First Bank Corporation')
and b.salary > 10000;
```

c.

```
select id
from employee a
where not exists(select 1 from works b
where a.id = b.id and b.company_name = 'First Bank Corporation');
```

d.

```
select id from employee
where salary > (select max(salary) from employee
where company_name = 'Small Bank Corporation');
```

e.

```sql
select distinct company_name from company a
where not exists(select 1 from company_name b where b.company_name = 'Small Bank Corporation'
and not exists(select 1 from company c where a.company_name = c.company_name
and b.city = c.city));
```

f.

```sql
with ta as
(select company_name,count(id) cnt
from works
group by company_name)
select company_name from ta
where cnt = (select max(cnt) from ta);
```

g.

```sql
with ta as
(select company_name,avg(salary) avg_sal
from works
group by company_name)
select company_name from ta
where avg_sal >
(select avg_sal from ta
where company_name='First Bank Corporation');
```

- 3.10

  a.

```sql
update employee set city = 'Newton' where id = '12345';
```

  b.

```sql
update works a set salary = case when salary <= 100000 then salary * 1.1
else salary * 1.03
end
where company_name = 'First Bank Corporation'
and exists(select 1 from managers b
where a.id = b.manager_id);
```

- 3.11

  a.

```sql
select name
from student a
```

```
where exists(select 1 from takes b natural join course c
where c.dept_name = 'Comp. Sci.'
and a.id = b.id);
```

b.

```
select id,name
from student a
where not exists(select 1 from takes b
where b.year < 2017
and a.id = b.id)
```

c.

```
select dept_name,max(salary)
from instructor
group by dept_name
```

d.

```
select min(max_sal)
from (
select max(salary) max_sal
from instructor
group by dept_name)
```

- 3.12

  a.

```
insert into course values('CS-001','Weekly Seminar',null,0);
```

b.

```
insert into section values('CS-001',1,'Fall',2017,null,null,null);
```

c.

```
insert into takes select id,'CS-001',1,'Fall',2017,null from student where
    dept_name = 'Comp. Sci.';
```

d.

```
delete from takes
where id = '12345' and year = 2017
and semester = 'Fall' and course_id = 'CS-001'
and sec_id = 1;
```

e.

- 如果 section 表的 course_id 没有建立外键，则删除可以进行。
- 如果建立了外键，如果没有定义外键的删除行为，那么删除会被阻止。
- 如果定义外键的删除行为为级联删除，那么删除会进行，并会把 section 表中的此课程的所有课程段都删除。
- 如果定位外键的删除行为为置空或者置缺省值，那么删除会进行，并会把 section 表中的此课程的所有课程段的 course_id 列设置成空值或缺省值。

f.

```
delete from takes a
where exists(select 1 from section b natural join course c
where a.course_id = course_id and a.sec_id = b.sec_id
and a.semester = b.semester and a.year = b.year
and lower(c.title) like '%advance%'
)
```

- 3.13

```
CREATE TABLE person
(
driver_id   INT PRIMARY KEY,
name        VARCHAR2 (20),
address     VARCHAR2 (30)
);

CREATE TABLE car
(
license   VARCHAR2 (10) PRIMARY KEY,
model     VARCHAR2 (20),
year      INT
);

CREATE TABLE accident
(
report_number   VARCHAR2 (10) PRIMARY KEY,
time            DATE,
location        VARCHAR2 (30)
);

CREATE TABLE owns
(
driver_id   INT PRIMARY KEY,
license     VARCHAR2 (10),
FOREIGN KEY (driver_id) REFERENCES person,
FOREIGN KEY (license) REFERENCES car
);

CREATE TABLE participated
(
report_number   VARCHAR2 (10),
license         VARCHAR2 (10),
driver_id       INT,
damage_amount   INT,
PRIMARY KEY (report_number, license),
FOREIGN KEY (report_number) REFERENCES accident,
FOREIGN KEY (driver_id) REFERENCES person,
FOREIGN KEY (license) REFERENCES car
);
```

- 3.14

  a.

```
select count(distinct report_number)
from person natural join owns join participated using(license)
where name = 'John Smith';
```

  b.

```
update participated set damage_amount = 3000
where report_number = 'AR2197'
and license = 'AABB2000';
```

- 3.15

  a.

```
select * from customer a
where not exists(
select 1 from branch b
where b.branch_city = 'Brooklyn'
and not exists(select 1 from account c natural join depositor d
where a.customer_name = d.customer_name
and b.branch_name = c.branch_name
)
)
```

  b.

```
select sum(amount) from loan;
```

c.

```
select branch_name
from branch
where assets > (select min(assets) from branch
where branch_city = 'Brooklyn')
```

- 3.16

  a.

```
select id,person_name
from employee a natural join works join company b using(company_name)
where a.city = b.city
```

b.

```
select id,person_name
from employee a
where exists (select 1 from employee b,managers c
where a.city = b.city
and a.street = b.street
and a.id = c.id
and b.id = c.manager_id)
```

c.

```
select id,person_name
from employee natural join works a
where salary > (select avg(salary) from works b
where a.company_name = b.company_name)
```

d.

```
with ta as (select company_name,sum(salary) sum_sal from works)
select company_name
from ta
where sum_sal = (select max(salary) from ta);
```

b.

```
select id,person_name
from employee a
where exists (select 1 from employee b,managers c
where a.city = b.city
and a.street = b.street
and a.id = c.id
and b.id = c.manager_id)
```

- 3.17

  a.

```sql
update works set salary = salary * 1.1
where company_name = 'First Bank Corporation';
```

  b.

```sql
update works a set salary = salary * 1.1
where company_name = 'First Bank Corporation'
and exists(select 1 from managers b where a.id = b.manager_id);
```

  c.

```sql
delete from works
where company_name = 'Small Bank Corporation';
```

- 3.18

```sql
create table employee(
id          int primary key,
person_name varchar(20),
street      varchar(20),
city        varchar(20)
);


create table company(
company_name varchar(20) primary key,
city varchar(20)
);

create table works(
id           int primary key,
company_name varchar(20),
salary       number(10,2)
FOREIGN KEY (id) REFERENCES employee,
FOREIGN KEY (company_name) REFERENCES company,
);

create table managers(
id           int primary key,
manager_id   int,
FOREIGN KEY (id) REFERENCES employee,
FOREIGN KEY (manager_id) REFERENCES employee
);
```

- 3.21

  a.

```sql
select name
from member a
where exists(select 1 from borrowed b natural join book c
where a.memb_no = b.memb_no and c.publisher = 'McGraw-Hill'
)
```

  b.

```sql
select name
from member a
where not exists(select 1 from book b
where b.publisher = 'McGraw-Hill'
and not exists(select 1 from borrowed c
where a.memb_no = c.memb_no
and b.isbn = c.isbn)
)
```

  c.

```sql
select name
from member natural join borrowed natural join book
group by memb_no,name,publisher
having count(distinct isbn) > 5
```

  d.

```sql
select avg(cnt)
from (
select count(*) cnt
from member natural left outer join borrowed
group by memb_no
)
```

- 3.22

```sql
where not exists(
select title
from course
group by title
having count(*) > 1)
```

- 3.23

```
SELECT dept_name
FROM instructor
GROUP BY dept_name
HAVING SUM (salary) >= (SELECT AVG (sum_sal)
FROM (  SELECT SUM (salary) sum_sal
FROM instructor
GROUP BY dept_name))
```

- 3.24

```
select id,name
from student a
where a.dept_name = 'Accounting'
and exists(select 1 from advisor b,instructor c
where b.iid = c.id and b.sid = a.id
and c.dept_name = 'Physics');
```

- 3.25

```
select depe_name
from department
where assets >
(select assets
from department
where dept_name = 'Philosophy')
order by 1
```

- 3.26

```
select course_id,id
from takes
group by course_id,id
having count(*) > 2;
order by 1
```

- 3.27

```
select id
from(
select course_id,id
from takes
group by course_id,id
having count(*) > 1
) group by id
```

```
having count(*) > 2
```

- 3.28

```
select id,name from instructor a
where not exists(
select 1 from course b
where a.dept_name = b.dept_name
and not exists(
select 1 from teachs c
where a.id=c.id
and c.course_id = b.course_id)
)
order by 2
```

- 3.29

```
select id,name
from student a
where dept_name = 'History'
and a.name like 'D%'
and (select count(distinct course_id)
from takes b natural join course c
where c.dept_name = 'Music'
and a.id = b.id) < 5
```

- 3.30

  只要有元组的salary为空，那么这个查询结果就不为 0。因为avg,sum计算时会忽略空值,而count(∗)是计算行数，salary为空对其没有影响。比如下列例子:

Salary
1000

  这时，avg计算结果为 1000，sum计算结果为 1000，count(∗)计算结果为 2，所以最终表达式的计算结果为 500

- 3.31

```
select id,name from instructor a
where not exists(
select 1 from (select * from takes  where grade = 'A')
join (select * from teachs c where c.id = a.id ) using(year,semester,
    course_id,sec_id)
)
```

- 3.32

```
select id,name from instructor a
where not exists(
select 1 from takes join teachs b using(year,semester,course_id,sec_id)
where grade ='A' and b.id = a.id)
)
and exists(
select 1 from takes join teachs c using(year,semester,course_id,sec_id)
where grade is not null and c.id = a.id)
)
```

- 3.33

```
select course_id,title
from course a
where exists(select 1 from section b natural join time_slot c
where a.course_id = b.course_id
and to_char(c.end_time,'hh24:mi' >= '12:00');
```

- 3.34

```
select course_id,sec_id,year,semester,count(*) num
from section natural join takes
group by course_id,sec_id,year,semester
```

- 3.35

```
with ta as (select course_id,sec_id,year,semester,count(*) num
from section natural join takes
group by course_id,sec_id,year,semester)
select * from ta
where num = (select max(num) from ta)
```