第三章作业

3.9

a.

```sql
select e.ID, e.person_name, e.city
from employee as e, works as w
where w.company_name = 'First Bank Corporation' and e.ID = w.ID;
```

b.

```sql
select ID, person_name, city
from employee
where ID in (select ID
            from works
            where w.company_name = 'First Bank Corporation'
                and w.salary > 10000);
```

c.

```sql
select ID
from employee
where ID not in (select ID
                from works
                where company_name = 'First Bank Corporation');
```

d.

```sql
select ID
from works
where salary > all (select salary
                    from works
                    where company_name = 'Small Bank Corporation');
```

e.

```sql
select distinct C1.company_name
from company as C1
where not exists ((select city
                 from company
                 where company_name = 'Small Bank Corporation')
                except
                 (select city
                 from company as C2
                 where C1.company_name = C2.company_name))
```

f.

```sql
select company_name
from works
group by company_name
having count(distinct ID) >= all(select count(distinct ID)
                                 from works
                                 group by company_name)
```

g.

```sql
select company_name
from works
group by company_name
having avg(salary) > (select avg(salary)
                     from works
                     where company_name = 'First Bank Corporation')
```

3.18

```sql
create table employee (
    ID int,
    person_name varchar(100),
    street varchar(100),
    city varchar(100),
    primary key (ID)
);

create table company (
    company_name varchar(100),
    city varchar(100),
    primary key (company_name)
);

create table works (
    ID int,
    company_name varchar(100),
    salary numeric(10, 2),
    primary key (ID),
    foreign key (ID) references employee on delete cascade,
    foreign key (company_name) references company on delete set null
);

create table manages (
    ID int,
    manager_id int,
    primary key (ID),
    foreign key (ID) references employee on delete cascade,
    foreign key (manager_id) references employee on delete set null
);
```