3.8

branch(branch name, branch city, assets)
customer (ID, customer name, customer street, customer city) loan (loan number, branch name, amount) borrower (ID, loan number) account (account number, branch name, balance) depositor (ID, account number)

Consider the bank database of Figure 3.18, where the primary keys are underlined. Construct the following SQL queries for this relational database.

a. Find the ID of each customer of the bank who has an account but not a loan.

```
select customer.id
from customer, depositor
where customer.id = depositor.id
except
select customer.id
from customer, borrow
where borrower.id = customer.id
```

不存在怎么弄？ 用except

b. Find the ID of each customer who lives on the same street and in the same city as customer '12345'.

等于另一个怎么弄？ 嵌套查询

```
select  id
from customer
where (street, city ) in ( select street,city
      from customer
      where id = '12345')
```

c. Find the name of each **branch** that has at least **one customer** who has an account in the bank and who lives in “Harrison”

大于等于1 怎么弄？ 自然连接连接上了就说明肯定有一个.

```
select distinct account.branch_name
from account, customer, depositor
where customer.id = depositor.id and  customer.customer_city = 'Harrison' and
account.account_number = depositor.account_number
```

3.9

employee (ID, person name, street, city) works (ID, company name, salary) company (company name, city) manages (ID, manager id) 都是第一个为主键

Consider the relational database of Figure 3.19, where the primary keys are underlined. Give an expression in SQL for each of the following queries.

a. Find the ID, name, and city of residence of each employee who works for “First Bank Corporation”.

```
select employee.id, employee.person_name,employee.city
from works,employee
where employee.id = works.id and company name = 'First Bank Corporation'
```

自然连接。

b. Find the ID, name, and city of residence of each employee who works for “First Bank Corporation” and earns more than $10000.

```
select employee.id, employee.name, employee.city
from works, employee
where works.id = employee.id and works.salary >= 10000 and works.company_name =
'First Bank Corporation
```

c. Find the ID of each employee who does not work for “First Bank Corporation”.

```
select employee.id from employee
Except
select employee.id from works, employee where works.id = employee.id and
works.company_name = 'First Bank Corporation'
```

d. Find the ID of each employee who earns more than every employee of “Small Bank Corporation”.

```
     SELECT id
         FROM works
          WHERE salary >= (SELECT max(salary)
                      FROM works
                               WHERE company_name ='Small Bank Corporation' )
```

having 是分组完成之后才能做的 ， where是分组完成之前的。

e. Assume that companies may be located in several cities. Find the name of each company that is located in every city in which “Small Bank Corporation” is located.

怎么保证每个都有?

如果有一个不在, 就去掉这个公司的名字.

```
SELECT company_name
FROM company
WHERE not exists (
SELECT city
FROM company
except
SELECT city
FROM company
WHERE company_name = 'Small Bank Corporation'
);
```

f. Find the name of the company that has the most employees (or companies, in the case where there is a tie for the most).

```
select company_name
from works
group by company_name
having count(id) >= (select count(id)
                     from works
                     group by company_name)
```

g. Find the name of each company whose employees earn a higher salary, on average, than the average salary at “First Bank Corporation”.

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

3.10

Consider the relational database of Figure 3.19. Give an expression in SQL for each of the following:

a. Modify the database so that the employee whose ID is '12345' now lives in “Newtown”.

```
update employee
set city = 'Newtown'
where id = '12345'
```

b. Give each manager of “First Bank Corporation” a 10 percent raise unless the salary becomes greater than $100000; in such cases, give only a 3 percent raise.

```
update works
set salary = case
            when salary> 100000 then salary*1.1
            else salary*1.03
            end
where company_name = 'First Bank Corporation'
```

3.11

Write the following queries in SQL, using the university schema.

a. Find the ID and name of each student who has taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.

b. Find the ID and name of each student who has not taken any course offered before 2017.

c. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.

d. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.

3.15

Consider the bank database of Figure 3.18, where the primary keys are underlined. Construct the following SQL queries for this relational database.

a. Find each customer who has an account at every branch located in “Brooklyn”.

b. Find the total sum of all loan amounts in the bank.

c. Find the names of all branches that have assets greater than those of at least one branch located in “Brooklyn”.