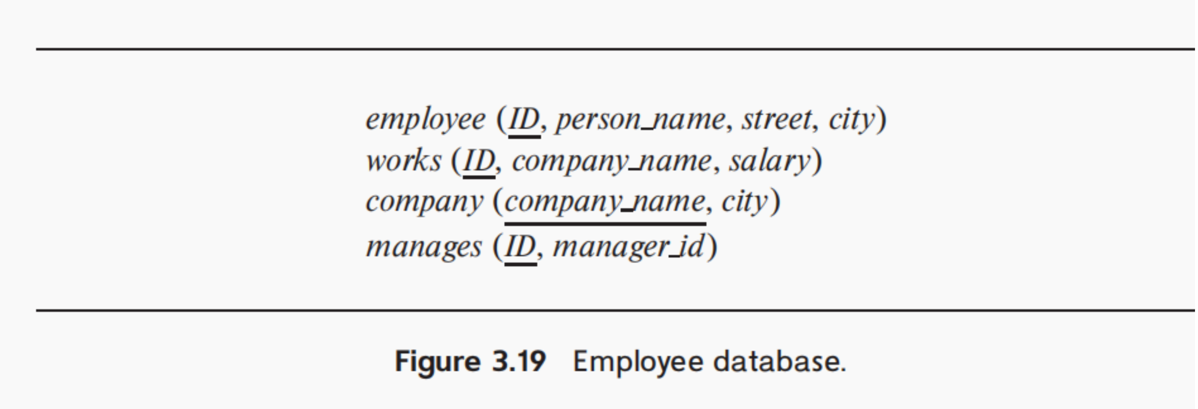
数据库作业week4

3.10

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



employee (*ID*, *person_name*, *street*, *city*)
works (*ID*, *company_name*, *salary*)
company (*company_name*, *city*)
manages (*ID*, *manager_id*)

Figure 3.19 Employee database.

> 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 as A
set A.salary =  case
                when A.salary>$100000 then A.salary*1.03
                else A.salary * 1.1
                end
where A.ID in (select ID
              from (works  natural join manages) as B
              where B.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.

```
select distinct student.ID,student.name
from student,course,takes
where student.ID = takes.ID and takes.course_id = course.course_id and dept_name
= 'Comp.Sci';
```

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

```
select ID,name
from student
where not exists( select *
                 from takes as A
                 where A.year<2017 and A.ID = student.ID
                    ) ;
```

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

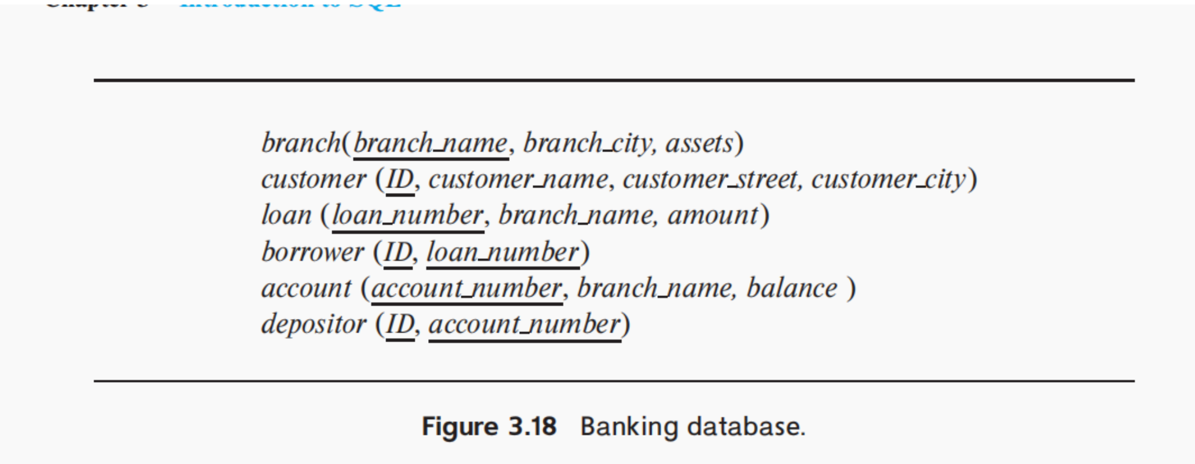
```
select dept_name,max(salary)
from instrutor
group by dept_name;
```

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

```
select min(A.max_salary)
from (  select dept_name,max(salary)
        from instrutor
        group by dept_name) as A(dept_name,max_salary);
```

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.



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*)

Figure 3.18 Banking database.

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

```
select A.ID,A.coustomer_name
from ((customer natural join depositor) natural join account) natural join branch
as A
where branch_city = 'Brooklyn'
group by A.ID
having count(distinct branch_name) = (  select count(distinct branch_name)
                                        from branch as B
                                        where B.branch_city = 'Brooklyn'
                                        );
```

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

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

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

```
select A.branch_name
from branch as A
where A.assets > some ( select B.assets
                        from branch as B
                       where B.branch_city = 'Brooklyn'
                      );
```