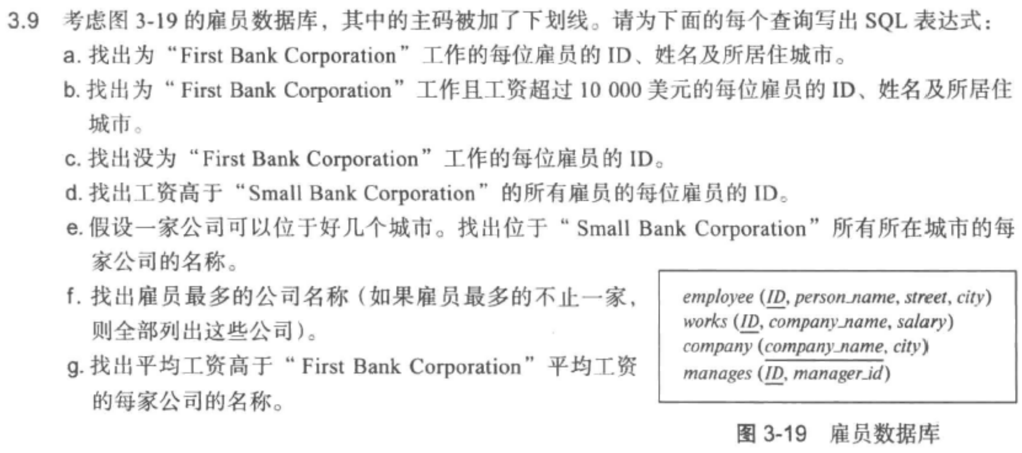
数据库第二次作业

2352018 刘彦



3.9 考虑图 3-19 的雇员数据库，其中的主码被加了下划线。请为下面的每个查询写出 SQL 表达式：

a. 找出为“First Bank Corporation”工作的每位雇员的 ID、姓名及所居住城市。

b. 找出为“First Bank Corporation”工作且工资超过 10 000 美元的每位雇员的 ID、姓名及所居住城市。

c. 找出没为“First Bank Corporation”工作的每位雇员的 ID。

d. 找出工资高于“Small Bank Corporation”的所有雇员的每位雇员的 ID。

e. 假设一家公司可以位于好几个城市。找出位于“Small Bank Corporation”所有所在城市的每家公司的名称。

f. 找出雇员最多的公司名称（如果雇员最多的不止一家，则全部列出这些公司）。

g. 找出平均工资高于“First Bank Corporation”平均工资的每家公司的名称。

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

图 3-19 雇员数据库

a.

```
select employee.ID, person_name, city
from employee, works
where employee.ID = works.ID and works.company_name = "First Bank Corporation"
```

b.

```
select employee.ID, person_name, city
from employee, works
where employee.ID = works.ID and works.company_name = "First Bank Corporation"
and works.salary > 10000
```

c.

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

d.

```
select distinct ID
from works
where salary > (
    select max(salary)
    from works
    where company_name = "Small Bank Corporation"
);
```

e.

①如果是查询至少位于“Small Bank Corporation”某个城市的公司

```
select distinct company_name
from company as c1
where exists (
    select *
    from company as c2
    where c2.city = c1.city and c2.company_name = "Small Bank Corporation"
);
```

②如果是查询其所有城市都在“Small Bank Corporation”所在城市范围内的公司

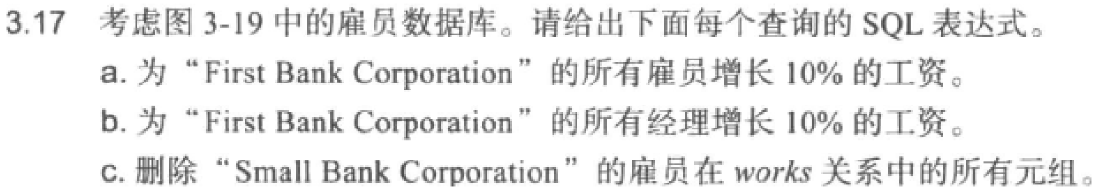
```
select distinct company_name
from company as c1
where not exists ((
    select city
    from company as c2
    where c2.company_name = c1.company_name)
    except (
        select city
        from company as c3
        where c3.company_name = "Small Bank Corporation")
);
```

f.

```
with CompantCount as (
    select company_name, count(distinct ID) as employee_count
    from works
    group by company_name
),
Maxcount as (
    select max(employee_count) as max_count
    from CompantCount
),
select distinct company_name
from CompantCount as cc
where exists (
    select *
    from Maxcount as mc
    where cc.employee_count = mc.max _count
);
```

g.

```
with FirstBankAvg as (
    select avg(salary) as avg_salary
    from works
    where company_name = "First Bank Corporation"
)
select company_name
from works
group by company_name
having avg(salary) > (
    select avg_salary
    from FirstBankAvg
);
```



3.17 考虑图 3-19 中的雇员数据库。请给出下面每个查询的 SQL 表达式。

a. 为“First Bank Corporation”的所有雇员增长 10% 的工资。

b. 为“First Bank Corporation”的所有经理增长 10% 的工资。

c. 删除“Small Bank Corporation”的雇员在 *works* 关系中的所有元组。

a.

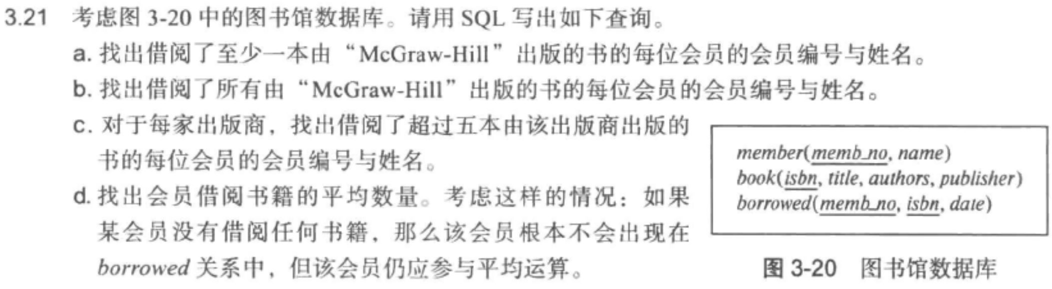
```
update works
set salary = salary × 1.10
where company_name = "First Bank Corporation"
```

b.

```
update works
set salary = salary × 1.10
where company_name = "First Bank Corporation" and exists (
    select *
    from manages
    where manages.manager_ID = works.ID
);
```

c.

```
delete from works
where company_name = "Small Bank Corporation"
```



3.21 考虑图 3-20 中的图书馆数据库。请用 SQL 写出如下查询。

a. 找出借阅了至少一本由“McGraw-Hill”出版的书的每位会员的会员编号与姓名。

b. 找出借阅了所有由“McGraw-Hill”出版的书的每位会员的会员编号与姓名。

c. 对于每家出版商，找出借阅了超过五本由该出版商出版的书的每位会员的会员编号与姓名。

d. 找出会员借阅书籍的平均数量。考虑这样的情况：如果某会员没有借阅任何书籍，那么该会员根本不会出现在 *borrowed* 关系中，但该会员仍应参与平均运算。

member(memb_no, name)
book(isbn, title, authors, publisher)
borrowed(memb_no, isbn, date)

图 3-20 图书馆数据库

a.

```
select distinct memb_no, name
from member
where exists (
    select *
    from borrowed
    where borrowed.memb_no = member.memb_no
    and exists (
        select *
        from book
        where book.isbn = borrowed.isbn and publisher = "McGraw– Hill")
);
```

b.

```
select distinct memb_no, name
from member
where not exists ((
    select isbn
    from book
    where publisher = "McGraw– Hill")
    except (
        select isbn
        from borrowed
        where borrowed.memb_no = member.memb_no)
);
```

C.

```
with PublisherBorrowers as  (
    select borrowed.memb_no, (
        select book.publisher
        from book
        where book.isbn = borrowed.isbn
    ) as publisher
    from borrowed
    group by borrowed.memb_no, (
        select book.publisher
        from book
        where book.isbn = borrowed.isbn
    )
    having count(*) > 5
)
select distinct memb_no, name
from member
where exists (
    select *
    from PublisherBorrowers as pb
    where pb.memb_no = member.memb_no
```

);

d.

```
select (sum((
        select count(∗)
        from borrowed
        where borrowed.memb_no = member.memb_no
)) × 1.0) / count(∗) as avg_borrow
from member
```