数据库系统原理第二次作业

2154312 郑博远

3.9 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".

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

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

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

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.

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).

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

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

Figure 3.19 Employee Database.

Solution:

a. ***select distinct*** *e.ID, person_name, city*
from *employee* ***as*** *e, works* ***as*** *w*
where *e.ID = w.ID* ***and*** *w.company_name = "First Bank Corporation"*

b. ***select distinct*** *e.ID, person_name, city*

```
from employee as e,  works as w
where e.ID = w.ID and w.company_name = “First Bank Corporation” and salary >
10000
```

c. *if one only works for a single company:*

```
select ID
from works
where  company_name ≠ “First Bank Corporation”
```

if one can work for several companies:

```
select ID
from employee
where not in
  (select ID
  from works
  where company_name = “First Bank Corporation”)
```

d. *if one only works for a single company:*

```
select ID
from works
where salary > all
  (select salary
  from works
  where company_name = “Small Bank Corporation”)
```

if one can work for several companies:

```
with employee_salary as
  (select ID, sum(salary) as tot_salary
  from works
  group by ID)
select ID
from employee_salary
```

```
    where tot_salary > all
      (select tot_salary
      from works as w, employee_salary as s
      where w.company_name = “Small Bank Corporation” and w.ID = s.ID)
e.  select c1.company_name
    from company as c1
    where not exist(
      (select city
      from company
      where company_name = “Small Bank Corporation”
      ) except
      (select city
      from company as c2
      where c2.company_name = c1.company_name)
     )
f.  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.  select company_name
    from works
    group by company_name
    having avg(salary) >
      (select avg(salary)
      from works
      where company_name = “First Bank Corporation”)
```

3.17 Consider the employee database of Figure 3.19. Give an expression in SQL for each of the following queries.

a. Give all employees of “First Bank Corporation” a 10 percent raise.

b. Give all managers of “First Bank Corporation” a 10 percent raise.

c. Delete all tuples in the works relation for employees of “Small Bank Corporation”.

Solution:

```
a.  update works
    set salary = salary * 1.1
    where company_name  = “First Bank Corporation”
b.  update works
    set salary = salary * 1.1
    where company_name  = “First Bank Corporation” and ID in
      (select manager_id
      from manages)
c.  delete from works
    where company_name = “Small Bank Corporation”
```

3.21 Consider the library database of Figure 3.20. Write the following queries in SQL.

a. Find the member number and name of each member who has borrowed at least one book published by “McGraw-Hill”.

b. Find the member number and name of each member who has borrowed every book published by “McGraw-Hill”.

c. For each publisher, find the member number and name of each member who has borrowed more than five books of that publisher.

d. Find the average number of books borrowed per member. Take into account that if a member does not borrow any books, then that member does not appear in the borrowed relation at all, but that member still counts in the average.

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

Figure 3.20 Library Database.

Solution:

```
a.  select m.memb_no, m.name
    from member as m
    where m.memb_no in(
            select memb_no
            from book natural join borrowed
            where publisher = “McGraw-Hill”
    )
b.  select m.memb_no, m.name
    from member as m
    where not exist(
           (select isbn
           from book
           where publisher = “McGraw-Hill” )except
           (select b.isbn
           from borrowed as b
           where b.memb_no = m.memb_no)
    )
c.  select publisher, memb_no, name
    from member natural join book natural join borrowed
    group by publisher, memb_no
    having count(isbn) > 5
```

```
d.  with book_per_member (memb_no, book_num) as
        (select memb_no, count(isbn)
        from member natural join borrowed
        group by memb_no)
    insert into book_per_member
        select memb_no, 0
        from member
        where memb_no not in
            (select memb_no
            from book_per_number)
    select avg(book_num)
    from book_per_member
```