db exercise 5

3200105872 庄毅非

5.6 Consider the bank database of Figure 5.21. Let us defifine a view *branch cust* as follows:Suppose that the view is *materialized* ; that is, the view is computed and stored. Write triggers to *maintain* the view, that is, to keep it up-to-date on insertions to *depositor* or *account* . It is not necessary to handle deletions or updates. Note that, for simplicity, we have not required the elimination of duplicates.

```
create view branch cust as
select branch name, customer name
from depositor, account
where depositor.account number = account.account number
```

Answer:

```
CREATE TRIGGER account_trigger after INSERT
ON `account`
FOR EACH row
begin
  INSERT INTO branch_cust
              (branch_name,
               customer)
  VALUES     (new.branch_name,
              (SELECT customer_name
               FROM   depositor
               WHERE  depositor.account_number = new.account_number));
end;

CREATE TRIGGER depositor_trigger after INSERT
ON `depositor`
FOR EACH row
begin
```

```
  INSERT INTO branch_cust
              (branch_name,
               customer)
  VALUES     ((SELECT branch_name
               FROM   account
               WHERE  account.account_number = new.account_number),
              new.customer_name);
end;
```

5.15 Consider an employee database with two relations where the primary keys are underlined. Write a function *avg salary* that takes a company name as an argument and fifinds the average salary of employees at that company. Then, write an SQL statement, using that function, to fifind companies whose employees earn a higher salary, on average, than the average salary at “First Bank”.

```
# definition of function avg_salary
drop function if exists avg_salary;
delimiter $
create function avg_salary( s_company_name varchar(20))
returns real
DETERMINISTIC
begin
    declare result real default 0;
    select avg(salary) into result from works where works.company_name = s_company_name;
    return result;
end $
delimiter ;

# find company whose average salary is larger than 'First_Bank'
select company_name from works
group by company_name
```

```sql
having avg(salary) > avg_salary('First Bank')
```

5.19 Suppose there are two relations *r* and *s* , such that the foreign key *B* of*r* references the primary key *A* of *s* . Describe how the trigger mechanism can be used to implement the on delete cascade option when a tuple is deleted from *s* .

Answer: 假设B表中的refering是A表中的refered属性的外键约束，我们可以设计如下的触发器。当我们删除A中的一个元组时，如果它在B中存在对应的元组，触发器保证首先删除B中对应的元组，在此之后我们删除A中的元组的时候，将不会受到外键约束的影响而造成删除失败。

```sql
create trigger cascadeDelete
    before delete on A
    for each row
    begin
        delete from B where B.refering = old.refered;
    end;
```