

5.6

Consider the bank database of Figure 5.21. Let us define a view branch_cust as follows:

```
create view branch_cust as
select branch_name, customer_name
from depositor, account
where depositor.account_number = account.account_number
```

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

答:

插入到 depositor, 如果有新customer_name, 更新,

插入到 account, 如果有新customer_name, 更新,

```
CREATE TRIGGER trig1 AFTER INSERT
ON depositor, account
referencing new row as nrow
for each row
begin
    insert into branch_cust
    (select branch_name, customer_name
     from account, depositor
     where depositor.account_number = nrow.account_number and
     depositor.account_number = account.account_number)
end
```

5.15

Consider the relational schema

part(part id, name, cost)

subpart(part id, subpart id, count)

where the primary-key attributes are underlined. A tuple (p1, p2, 3) in the subpart relation denotes that the part with part id p2 is a direct subpart of the part with part id p1, and p1 has 3 copies of p2. Note that p2 may itself have further subparts. Write a recursive SQL query that outputs the names of all subparts of the part with part-id 'P-100'.

返回一个平均工资, 就是标值.

```
create function avg_salary(@name varchar(10))
returns table
as
return select avg(salary) from works where company_name=@name
```

```

with eachsalary (company_name, eachsalary) as
  select company_name, avg_salary(company_name)
  from works
  group by company_name
select company_name
from eachsalary
where eachsalary > all 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*.

```

CREATE TRIGGER trig1 AFTER delete
ON s
referencing old row as orow
for each row
begin
  delete from r
  where orow.A = r.B
end

```