5.6

```
create trigger insert_to_depositor
after insert on deposior
as
insert into branch_cust
    select branch_name, customer_name
    from   new , account
    where  new . account_number = account . account_number

create trigger insert_to_account
after insert on accout
as
insert into branch_cust
    select branch_name, customer_name
    from   depositor, new
    where  depositor.account_number = new . account_number
```

5.15

```
create function avg_salary (company_n varchar(15))
  returns integer
  begin
  declare a  integer;
      select avg(salary) into result
      from   works
      where  works . company-name = company_n
  return a
  end

  select company_name
  from   works
  where  avg_salary(company_name) > avg_salary("First Bank Corporation")
```

补充

```
create precedure   insert_class ( in ID varchar(10), course_id varchar(10),
                                  sec_id varchar(10), semester varchar(10),
                                  year int, grade int)

    begin


    select time_slot_id, day, start_time as cur_time
    from   takes natural join section natural join time_slot
    where  takes . ID = ID

    declare pre  varchar(10)
    select prereq_id as cur_pre
    from   prereq
    where  course_id = prereq.course_id


   declare cur_cap int
   select capacity into cur_cap
   from  takes natural join section natural join classroom
   where takes.ID = ID and takes . course_id = course_id
```

```
if((select time.slot.id, day, start_time
    from takes natural join section natural join time_slot
    where course.id = takes.course_id and sec_id =
          takes.sec_id and semester = takes.semester
          and year = takes . year) in cur_time)
    then set errorMsg= "time conflict";
else if ( cur_pre not all in (select course_id
                              from  takes
                              where ID = takes.ID)
    then set errorMsg = "prerequest course
                         not supply";
else if ( capacity ≤ 0 )
    then set errorMsg= "Not enough capacity";
else (insert into takes
      [ID, course_id, sec_id, semester, year, grade];
 end if
end;
```