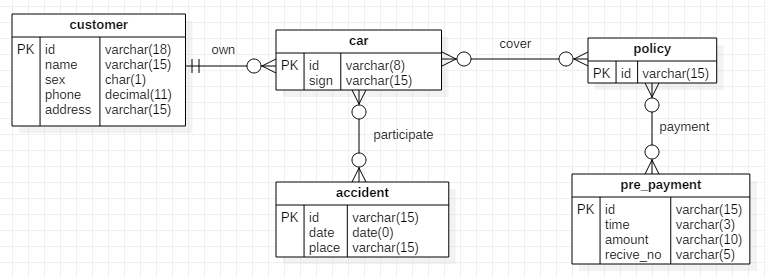
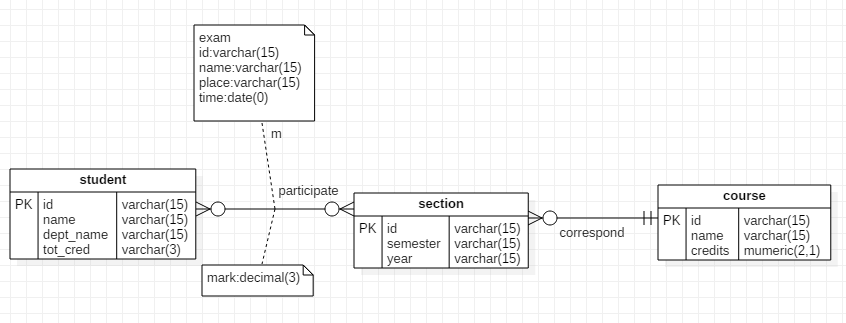
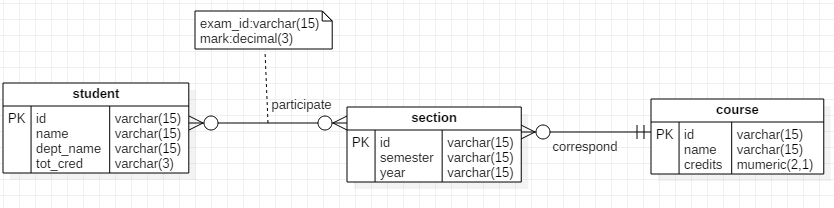
1. 注：以下er图是用专门软件（staruml）画的，软件没有提供双线。

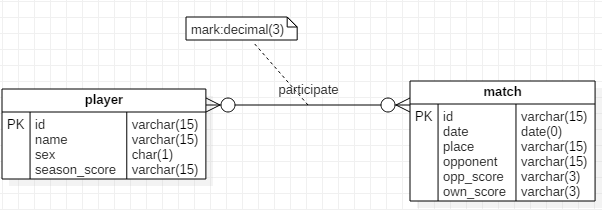


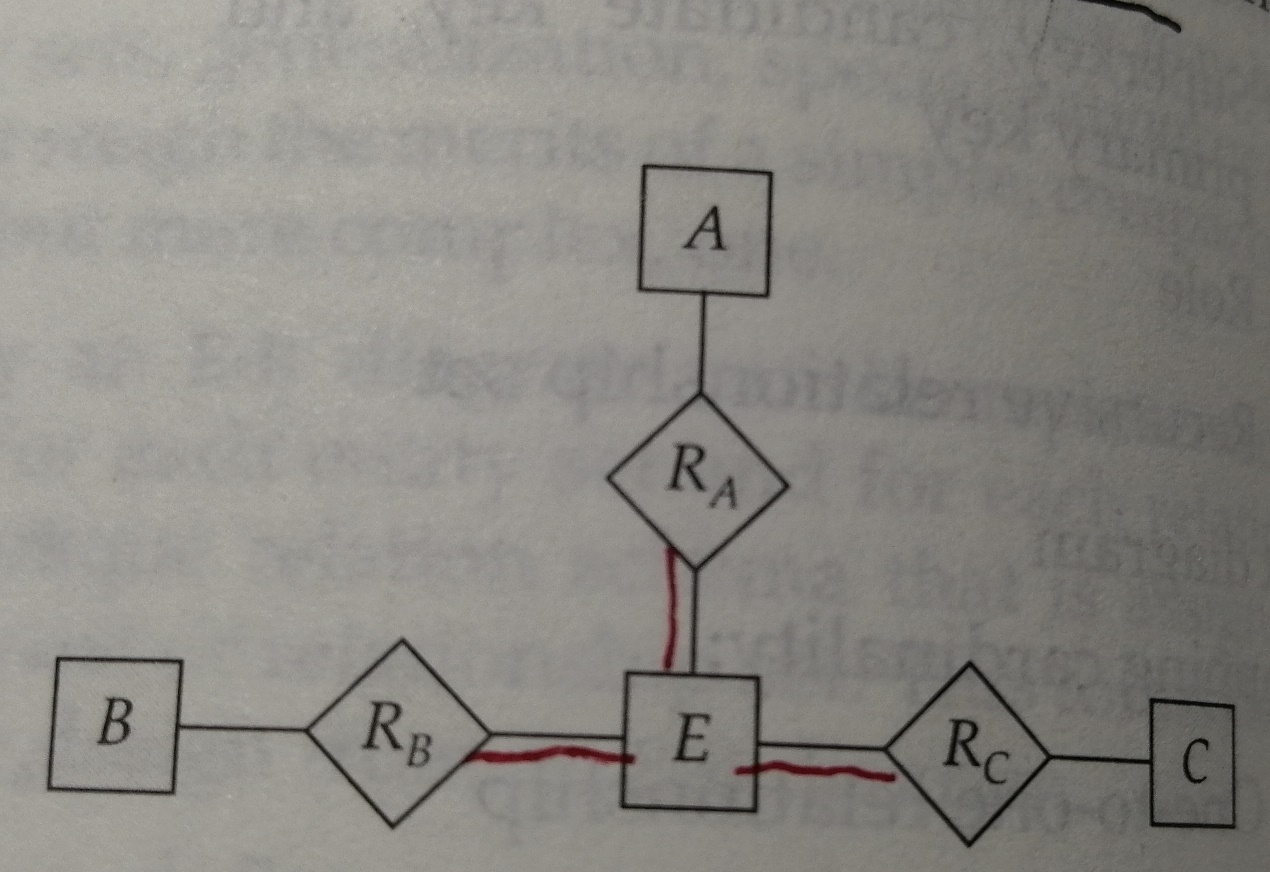
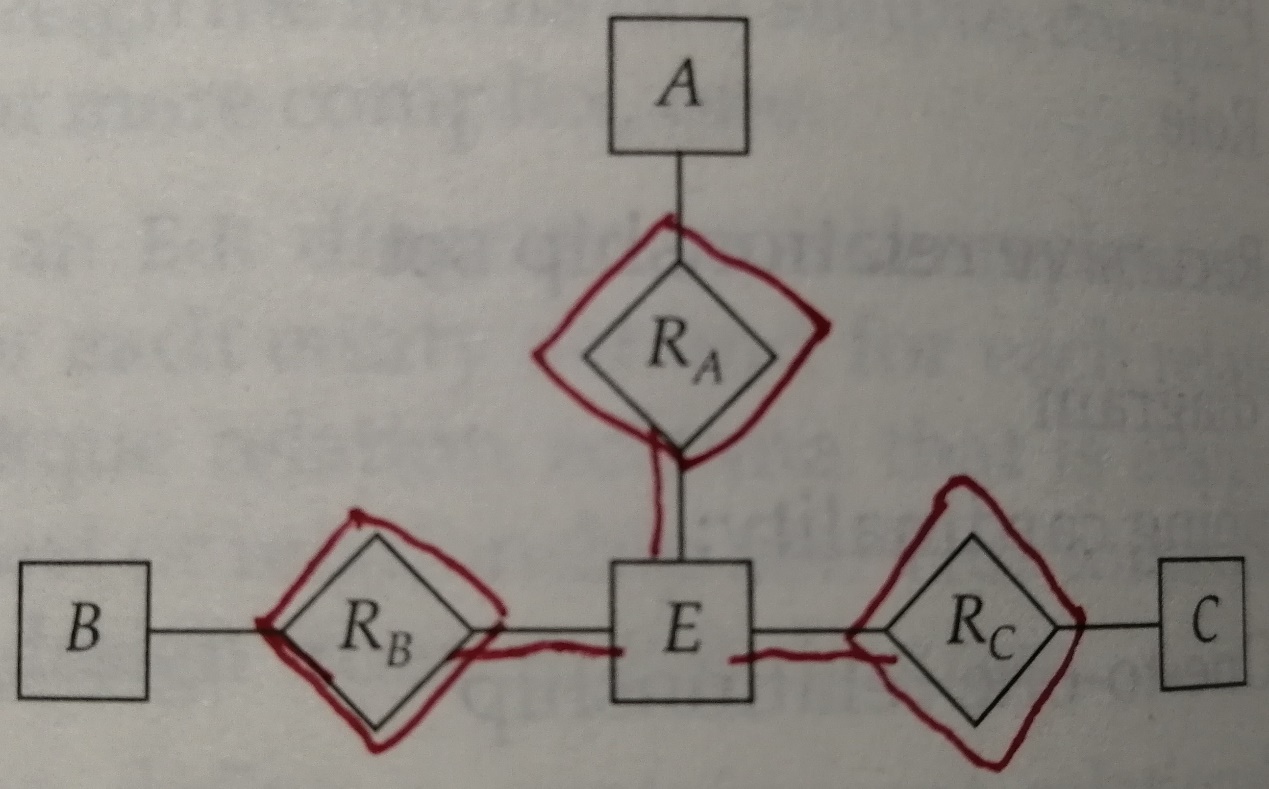
2. 那个软件画不出三元关系和联系上属性，所以用备注代替了一个实体，如下图：









1. 略
2. 略
4. 
5. 如果A完全参与关系R，那么就引入A和RA之间的总参与约束。
6. 
7. 略

section的主键是course\_id、sec\_id、semester、year组成，course\_id是参照course表的外键，这些约束使得多对一的基数约束强制执行，而全参与约束则不能强制执行，因为一些课程可能不会参与该关系。

1. 略

R中对应的主键约束b不能为空，这样就保证了R中所有元组都与B相关，所以A全参与了。举个例子，a是A中没有对应条目的元组，与r结合时，b为空，而这是不允许的。，

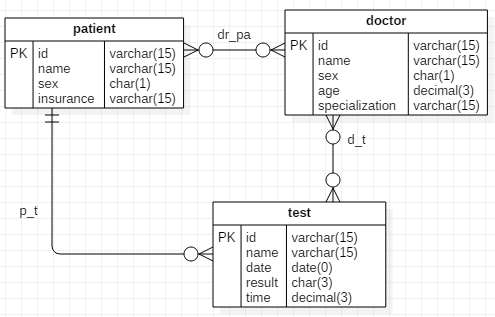
1. 略
2. 略
3. 略

主码：一个候选码；

候选码：最少字段的超码；

超码：唯一确定元组的某些或某个字段。







7.1 customer(id,name,sex,phone,address)

car(id,cus\_id,sign)

accident(id,date,place)

participate(car\_id,acc\_id)

policy(id)

cover(car\_id,po\_id)

pre\_payment(id,time,amount,recive\_no)

payment(po\_id,pre\_id)

7.2 student(id,name,dept\_name,tot\_cred)

section(id,course\_id,semester,year)

participate(exam\_id,stu\_id,sec\_id,mark)

course(id,name,credits)

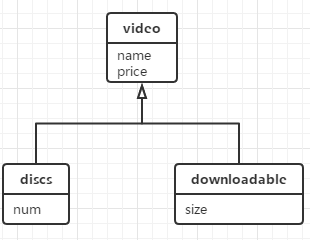
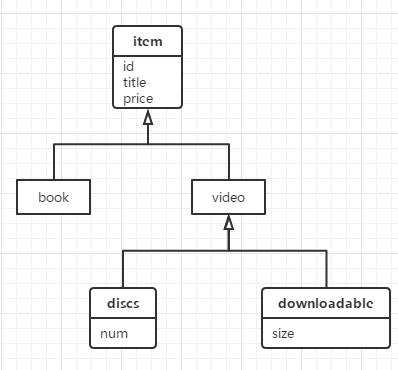
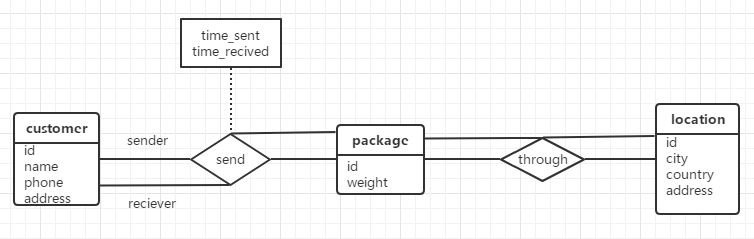
7.3 player(id,name,sex,season\_score)

match(id,date,place,opponent,opp\_score,own\_score)

participate(pla\_id,mat\_id,mark)

1. 略

强实体集具有主键，能根据它确定唯一元组。弱实体集本身没有主键，当包含所依赖的强实体集的属性，才能区分每一个元组。

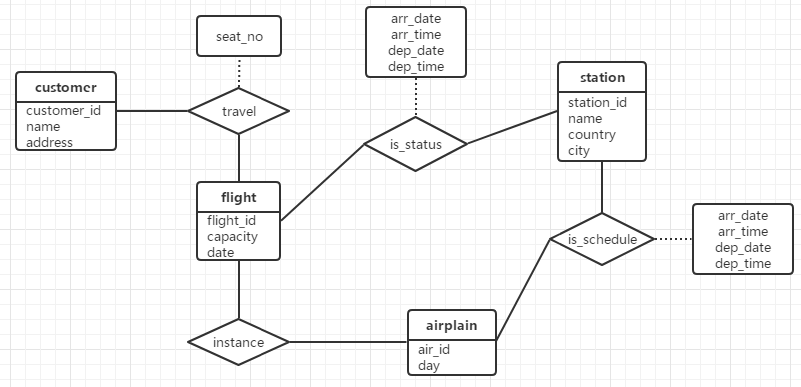
2. 避免数据重复和由此产生的可能的不一致（由复制强实体的键引起）
3. 弱实体反映了依赖实体的逻辑结构
4. 强实体被删除时，弱实体就会跟着被删
5. 弱实体可以与强实体的物理存储在一起
7. author(name,address,URL)  
   book(ISBN,title,year,price)  
   publisher(name,address,phone,URL)  
   shopping\_basket(basket\_id)  
   warehouse(code,address,phone)  
   customer(email,name,address,phone)
8. 
9. 
10. 略
11. //上图下滑线不好加  
    customer(id,name,phone,address)  
      
    package(id,weight)  
      
    location(id,city,country,address)  
      
    through(p\_id,l\_id,  
    foreign key p\_id references package,

foreign key l\_id references location)  
  
send(s\_id,r\_id,p\_id,time\_sent,time\_recived,  
foreign key s\_id references customer,

foreign key r\_id references customer,

foreign key p\_d references packet)



  
customer(customer\_id,name,address)  
  
flight(flight\_id,capacity,date)  
  
airplain(air\_id,day)  
  
station(station\_id,name,country,city)  
  
travel(flight\_id,customer\_id,seat\_no,  
foreign key flight\_id references flight,

foreign key customer\_id references customer)  
  
instance(flight\_id,air\_id,  
foreign key flight\_id references flight,  
foreign key air\_id references airplain)  
  
is\_schedule(air\_id,station\_id,arr\_date,arr\_time,dep\_date,dep\_time,   
foreign key air\_id references airplain,  
foreign key station\_id references station)

is\_status(flight\_id,station\_id, arr\_date,arr\_time,dep\_date,dep\_time,  
foreign key station\_id references station,  
foreign key flight\_id references flight)