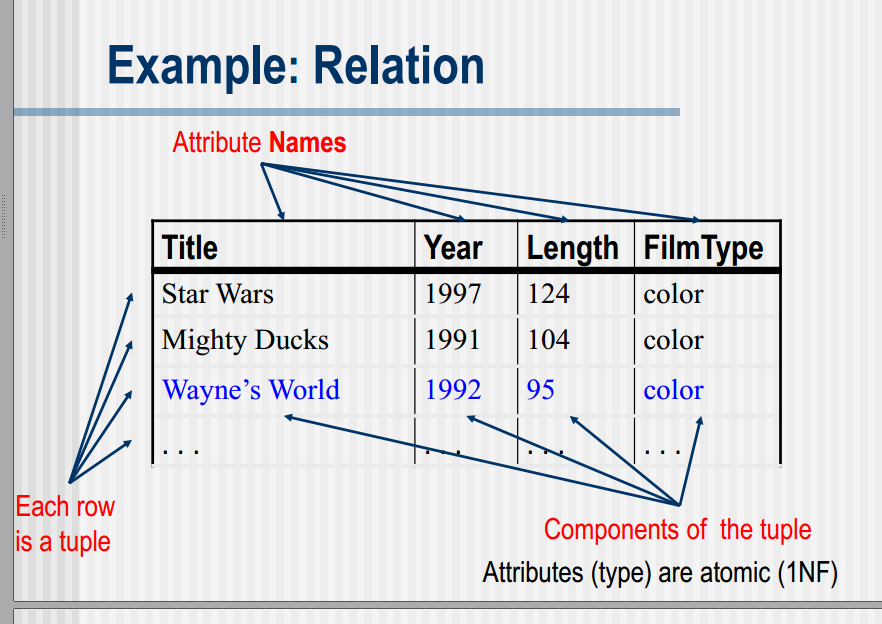
重点： 这章relation/table指的是entity set， relationship才是链接

Relational database

一组relations

啥叫relation

一个2-Dtable，管理DATA怎么放



这就是个标准relation

每一行row都是一个tuple

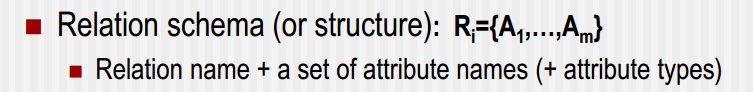
每一列都是attribute

attribute are atomic，只能有一个值

RelationSchema

一个Relation（也就是一个表格）

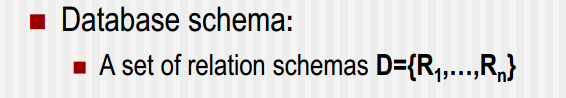
的结构是R={A1,A2,...AM} R是表格的名字



Relation Instance: 当前**时间点**所有表格tuples的值

Database Schema

一组Relation Schemas （表格名字组成总的database）

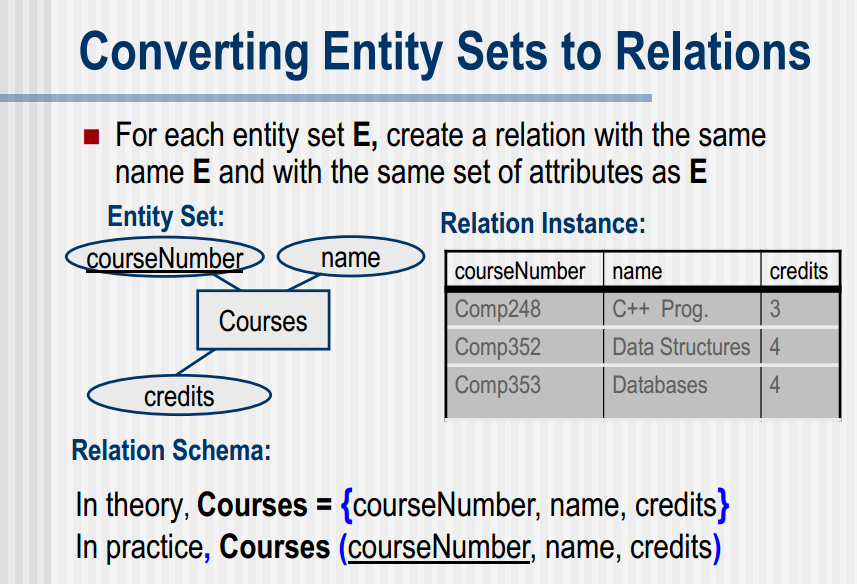


Database instance:

所有relation instances的集合

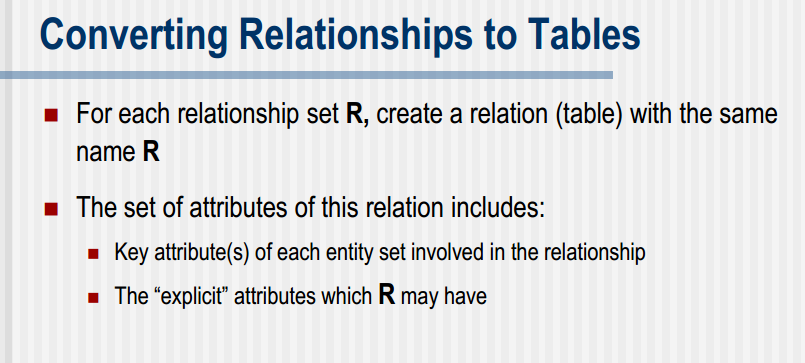
把一个ER图作为输入，输出一个relational database schema

普通Entity转换



把方框作为名字，别的作为attribute,横线作为key

Relation Set转换

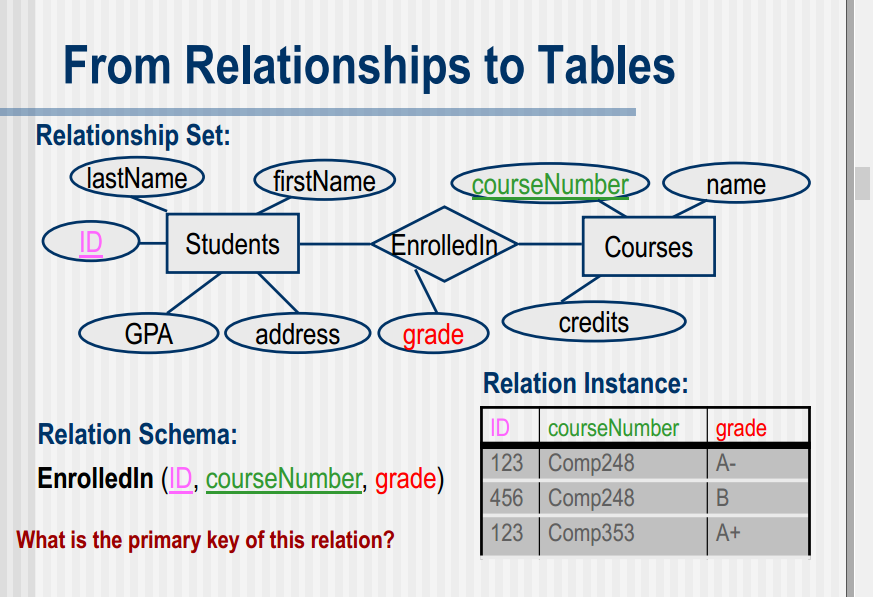


对于每个relation Set，

创造一个table，名字相同

这个table的attribute为 所有相连的entity set的key attribute

还有这个R本身带的explicit attribute



所连的attribute的key attribute就是你的key attribute

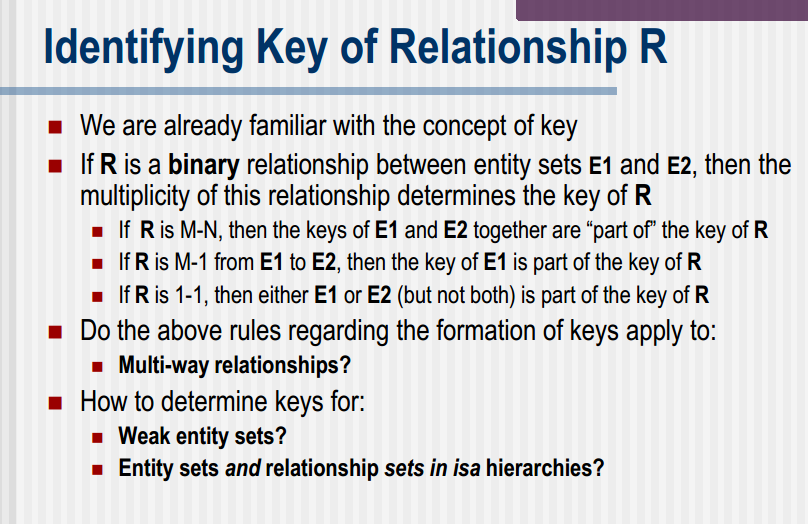
你的Key就是（id,coursenumber）

如果R是一个binary relationship(只连接两个entity set) 那么relationship 的multiplicity决定了R的key

如果R是M-N，那么 E1 E2的key组成了R的key

如果R是M-1, 那么M那头的key作为R的key的一部分

如果R是1-1，那么E1或E2都能组成R的key的一部分

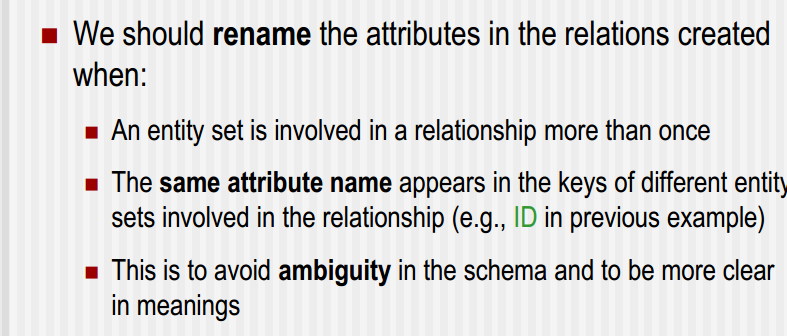


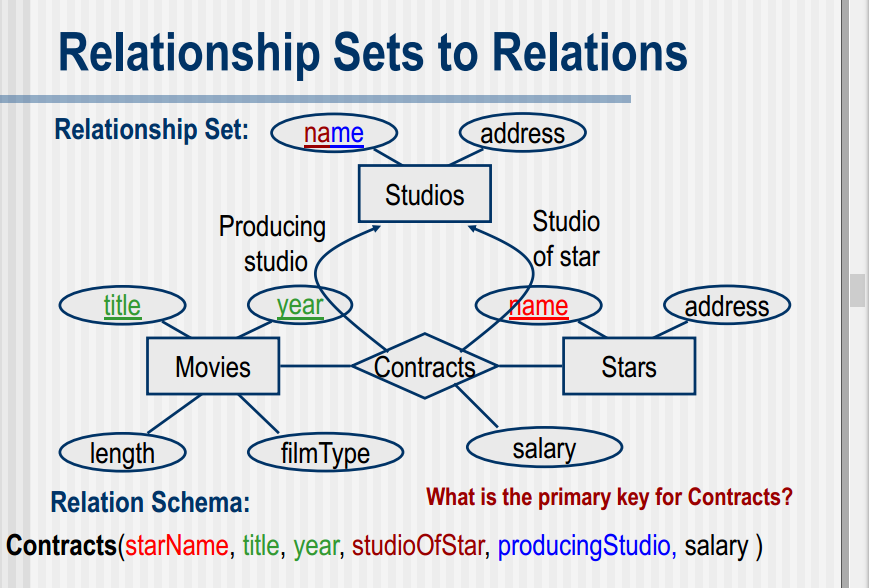
我们需要对ATTRIBUTE重新命名当

relashionship中的 某个entity set出现了大于一次

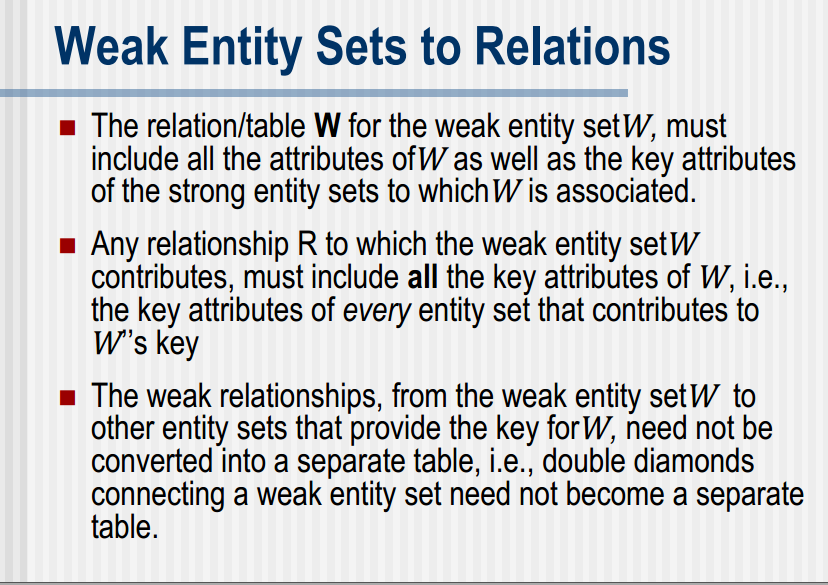
一个relationship中,不同的entity set有着同样的name的attribute

避免ambiguity





Contract 的primary key是starName,title,year,studioName

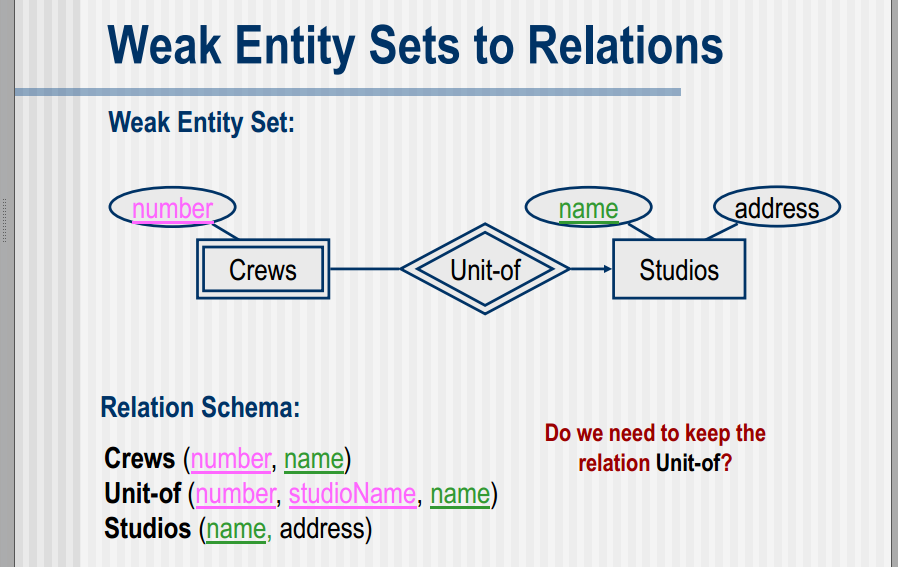


Weak Entity Set

Weak Entity本身的表，在除了它本身元素以外，还要加上他所连接的strong entity set的key attributes

而对于relashion Ship, 要包含所有weak Entity set的key attribute，以及其他与这个relationship相连的正常entity的key

但是从一个weak entity set连接到其他set ，而这个set实际上给这个weak entity set提供了key以后，就不需要转换成独立table ，因为weak entity set 已经描述了他爹是谁



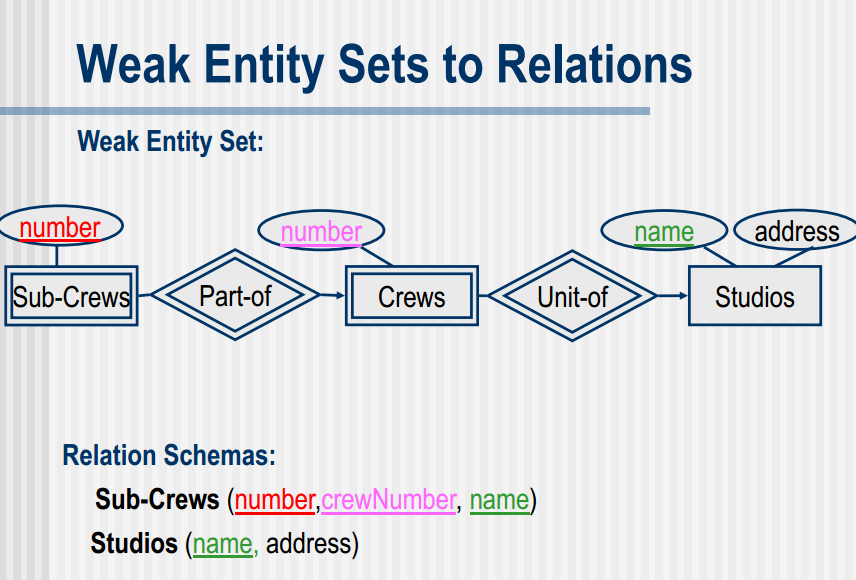
你可以发现Crew里面已经指明了他的NTO1中的1是谁

而如果要写relation, 那就需要加上weak entity中所有 key 也就是number+studioName, 然后再加上正常连的key，也就是name

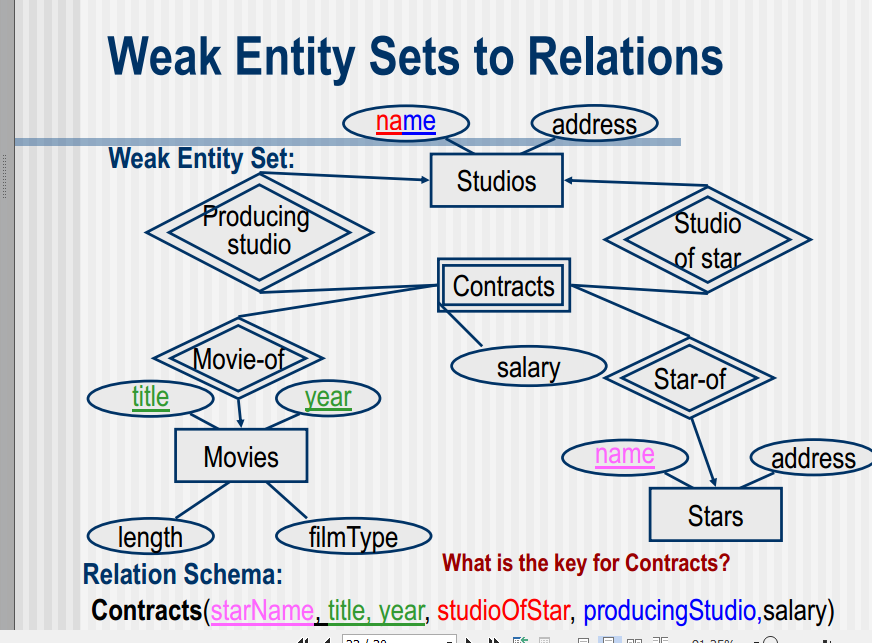
那么studioName实际上合NAME是一个东西

因此不需要，也就是第三条，没必要用UNIT-OF

但是从一个weak entity set连接到其他set ，而这个set实际上给这个weak entity set提供了key以后，就不需要转换成独立table ，因为weak entity set 已经描述了他爹是谁



连续的这种，他要包含一直到Strong entity为止所有的key



把isa继承链接转换成RelatioNs

三种方法：

1.Straight-ER style method

在ER模型里， 一个OBJECT实际上是由多个entity sets 中提取的entity组成的，然后通过isa 来连接

这些连接起来的entity决定了object的属性

2.Object-Oriented method

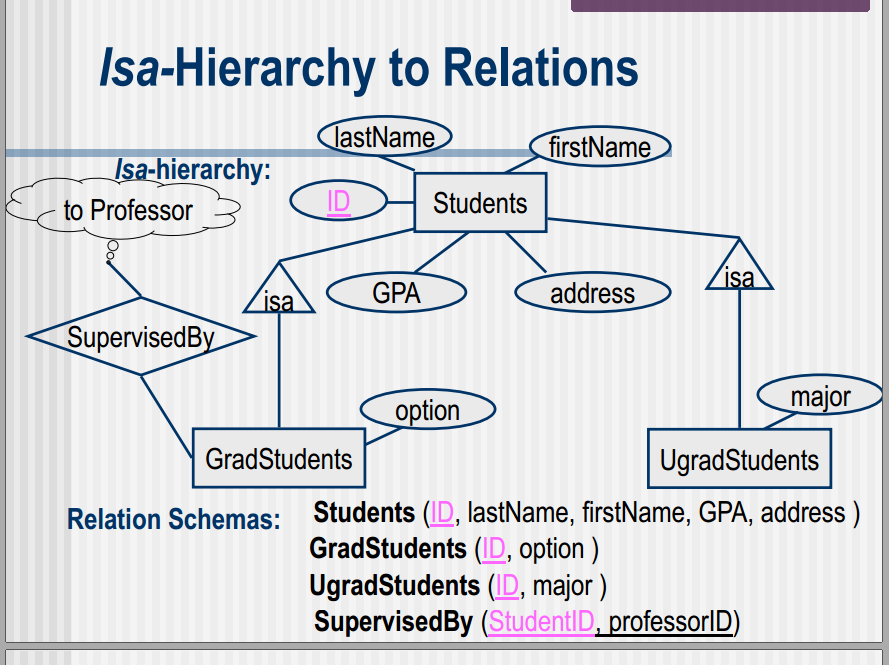
3.null method

对于每一个entity set实体，创造一个relation(table)

里面的attributre：属于他的attribute

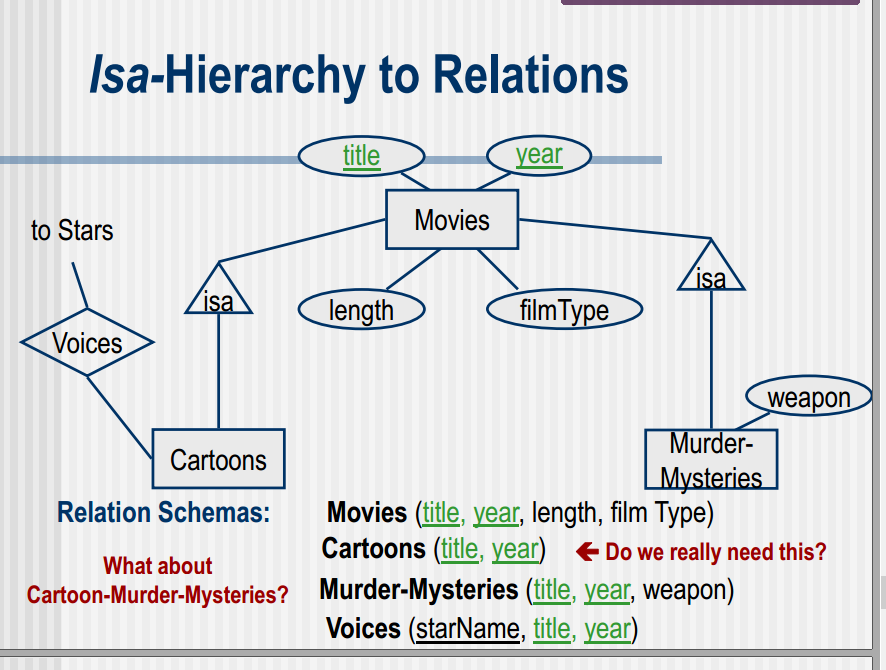
2.parrent的key attribute

isa 这个点不会创造relation

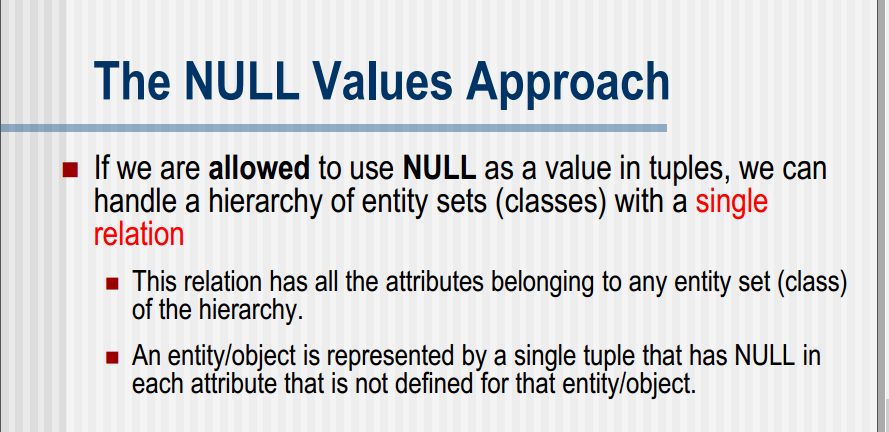


GradStudent没有写过多的内容，而只是有ID也就是父类的key，以及他自己的option

到时候ID相等然后链接表格就行

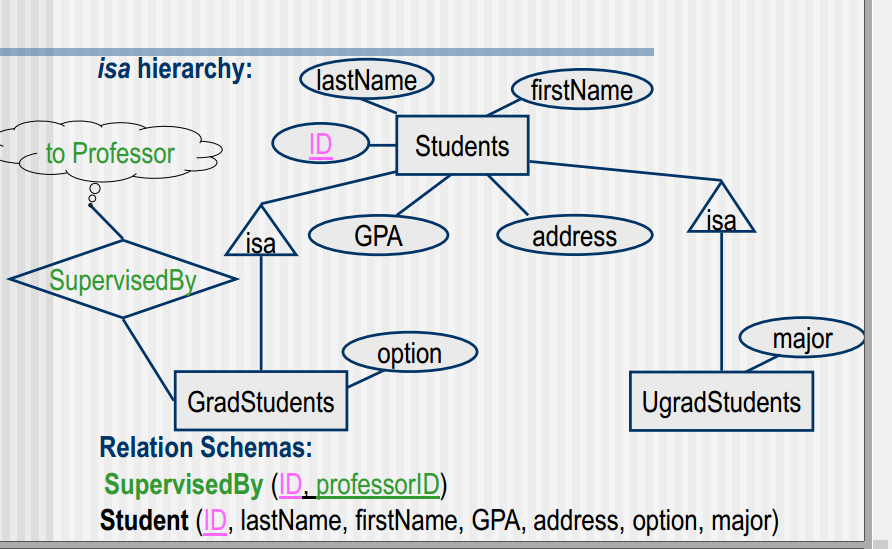


我们这里并不需要为Cartoon创造表格，因为他没有额外属性

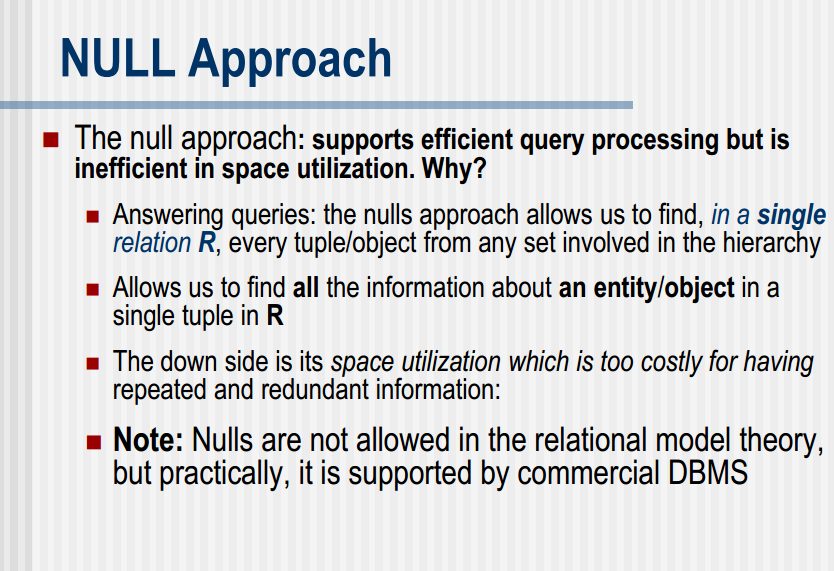


如果我们允许null作为tuple的attribute的值，

那么直接用原来的父类relation，把属于她的所有子类的attribute加上，例如graduate student，他的额外属性就是option，那么undergraduate option填Null就完事儿了



把gradStudent,ugradStudent的信息融入了Student,不是就填null



空间利用效率低

null approach允许在一个relation R(注意是relation ,而不是relationShip) 把他所有的子类都表示出来

但是space utilization太高，因为有重复的信息

在理论中不允许null，但是商业DBMS中支持