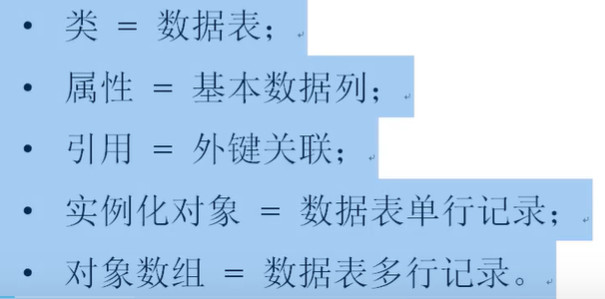
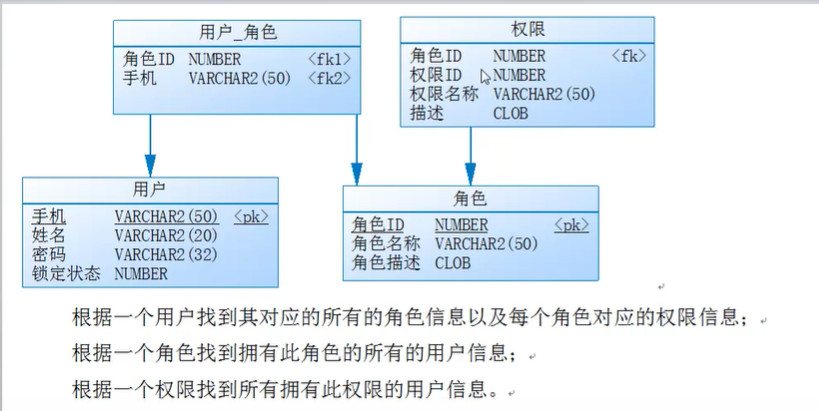
简单Java类就是基本能与数据表进行对应的的实体类。





将数据表转换为类，只需要写正常的类进行不需要关注关联关系。

String的两种实例化方法，直接赋值，或者使用构造方法。

首先比较分为两种类型：

数值型比较、非数值型比较。

数值型比较：比如说：字符、数值、引用（地址非实体）。

非数值型比较：就是引用的实体进行比较。（那么必须要你自己指定比较的原则，而因为String比较特殊所以Java进行了一个实现，可以使用equals进行比较。）

字符串的比较：如果直接将两个引用使用算术运算符进行比较，那么比较的是两者的地址，也就是比较的是引用而不是引用的实体。

所以字符串比较需要使用.equals()进行比较。

字符串常量：是一个匿名对象。

常量的引用不能改变常量所在内存中的值，如果改变，仅仅只是将引用的指向改变。

Hotspost虚拟机有常量池的概念。

而非常量的引用，是分配在堆区中的，所以改变其值是可以直接修改内存中的值，当然也可以改变引用的地址。

空对象，不可操作方法。

String的intern()方法：

尽管在输出中调用intern方法并没有什么效果，但是实际上后台这个方法会做一系列的动作和操作。在调用”ab”.intern()方法的时候会返回”ab”，但是这个方法会首先检查字符串池中是否有”ab”这个字符串，如果存在则返回这个字符串的引用，否则就将这个字符串添加到字符串池中，然会返回这个字符串的引用。

字符串常量池：

利用直接赋值形式的所有字符串都保存在一个对象池中，对象池就相当于一个字符串数组。每次进行新的字符串常量定义时，都会查找该常量池。

静态常量池:编译得到的.class文件作用之一就是用于对程序的内存进行一个先期分配以及对代码的优化，所有的内容都以常量的形式出现，并且都会对其资源分配完毕。就是死代码，值已经确定了的，不需要运行时计算才能够确定，那我现在就直接用这个值直接进行运算。

动态常量池：所有的内容不是固定的，就是有变量，而这个变量在编译时仅仅只是符号引用，所以需要在运行时才能够进行确定，即便你的变量被初始化了一个常量值，但是用来参与运算。

通过+的字符串的只是修改引用空间，会产生大量的垃圾，所以要慎用。