# Hibernate学习笔记

* hbm.xml文件中配置的<property name=”age” type=”int” column=”age”/>中的name对应的是pojo中的getter和setter方法，与pojo中的attribute（属性）无关。

其中type中的类型可以使用hibernate类型，也可以使用Java中的类型，但是最终还是会换换成hibernate中的类型。

* 在hibernate的project/etc文件夹中有一个hibernate.properties文件，里面包含了所有的东西如何配置的信息。
* 连接池的配置方法
* hibernate会使用默认的内置连接池，但不是最优的。
* 直接在hibernate.cfg.xml中直接配置c3p0的相关信息
* 在hibernate.cfg.xml中配置JNDI(在Tomcat中配置的JNDI)

**sessionFactory与session的特点**

一、创建SessionFactory

SessionFactory sessionFactory = conf.buildSessionFactory();

1.重量级对象 创建消耗资源大 耗时长

不建议频繁的创建与关闭

一个数据库创建一个SessionFactory

2.线程安全

3.缓存根据配置文件生成的CRUD语句

4.维护二级缓存

5.创建Session对象

二、创建Session对象

Session session = sessionFactory.openSession();

1.负责执行CRUD操作 session.save(topic);

2.轻量级对象 可以频繁的创建与销毁

3.线程不安全 建议一个线程独享一个Session

一个事务对应一个Session的使用

4.维护一级缓存

三、Session常用方法

1.save(Object obj) 保存对象 并返回主键

2.persist(Object obj) 保存对象无返回值

3.get(Class c,主键) 根据主键 读取对象

如果对应的记录不存在 返回空

4.load(Class c,主键) 根据主键 读取对象

采用懒加载机制

5.update(Object obj) 更新不被Session管理的对象

更新结束后，将对象重新加入Session管理

6.delete(Object obj) 删除对象，如果该对象在Session管理范围内，则在Session缓存中移除该对象

三、session的优化

1、使用threadLocal<session> local = new ThreadLocal<session>()

进行session和当前线程对象绑定，防止线程不安全。需要手动的关闭session。

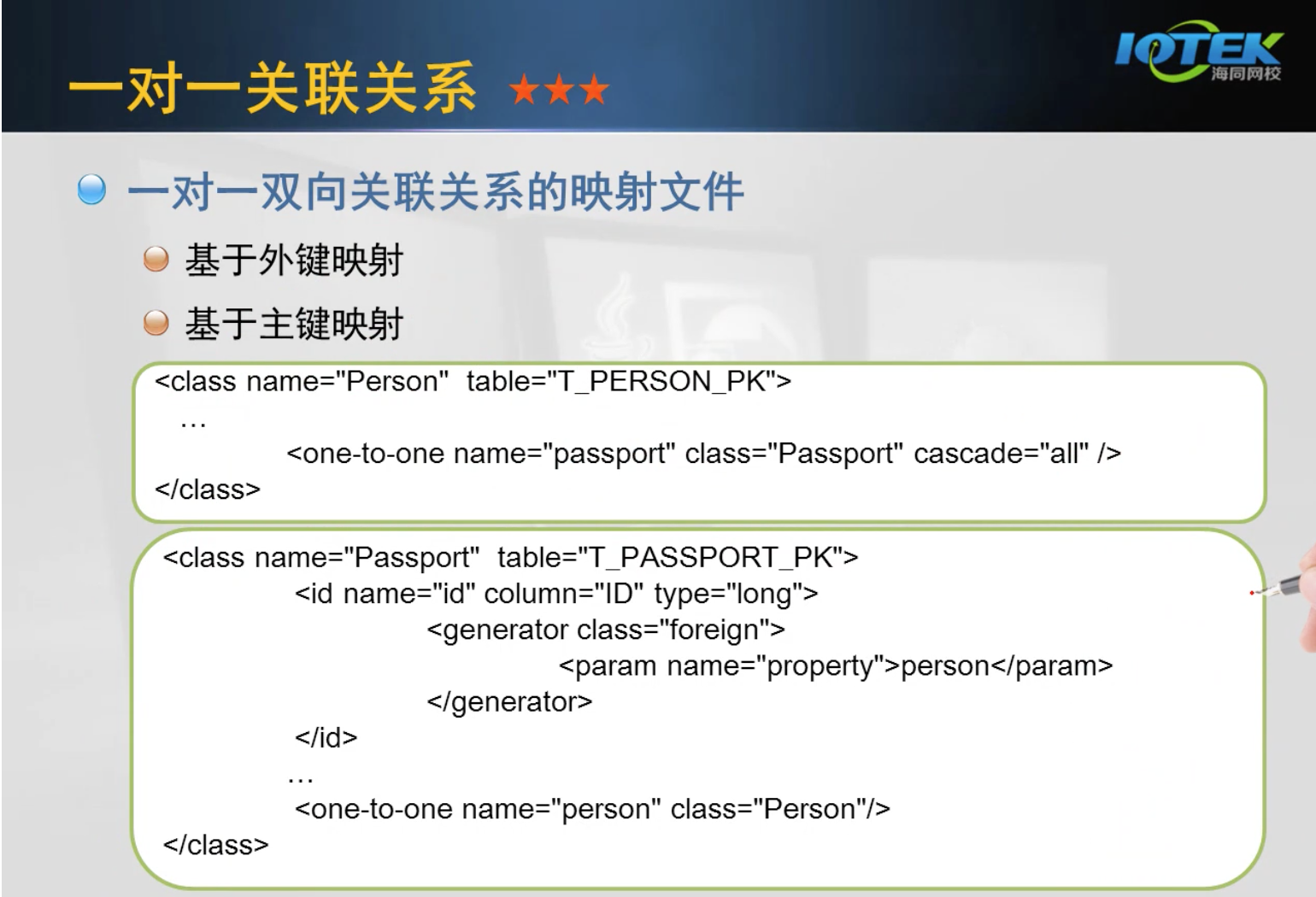
2、sessionFactory.getCurrentSession()直接获取session，内部自动维护session的开启和关闭。推荐使用。

四、一对一映射的实现方式

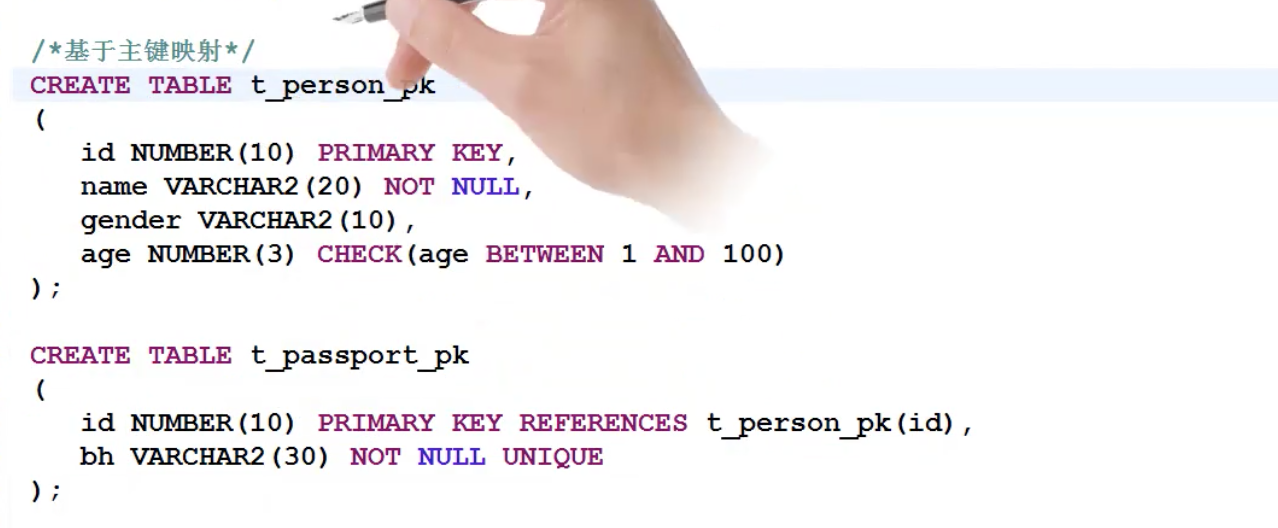
一对一的映射关系，指一个对象和另一个对象是一对一的关系，在类中的表现为，各自分别拥有对方类的一个实例作为自己类的成员变量，形成一种相互关联的关系。在数据库中的表现为如下两种方式：

1. 基于主键的映射

就是指在字表中外键作为字表的主键。

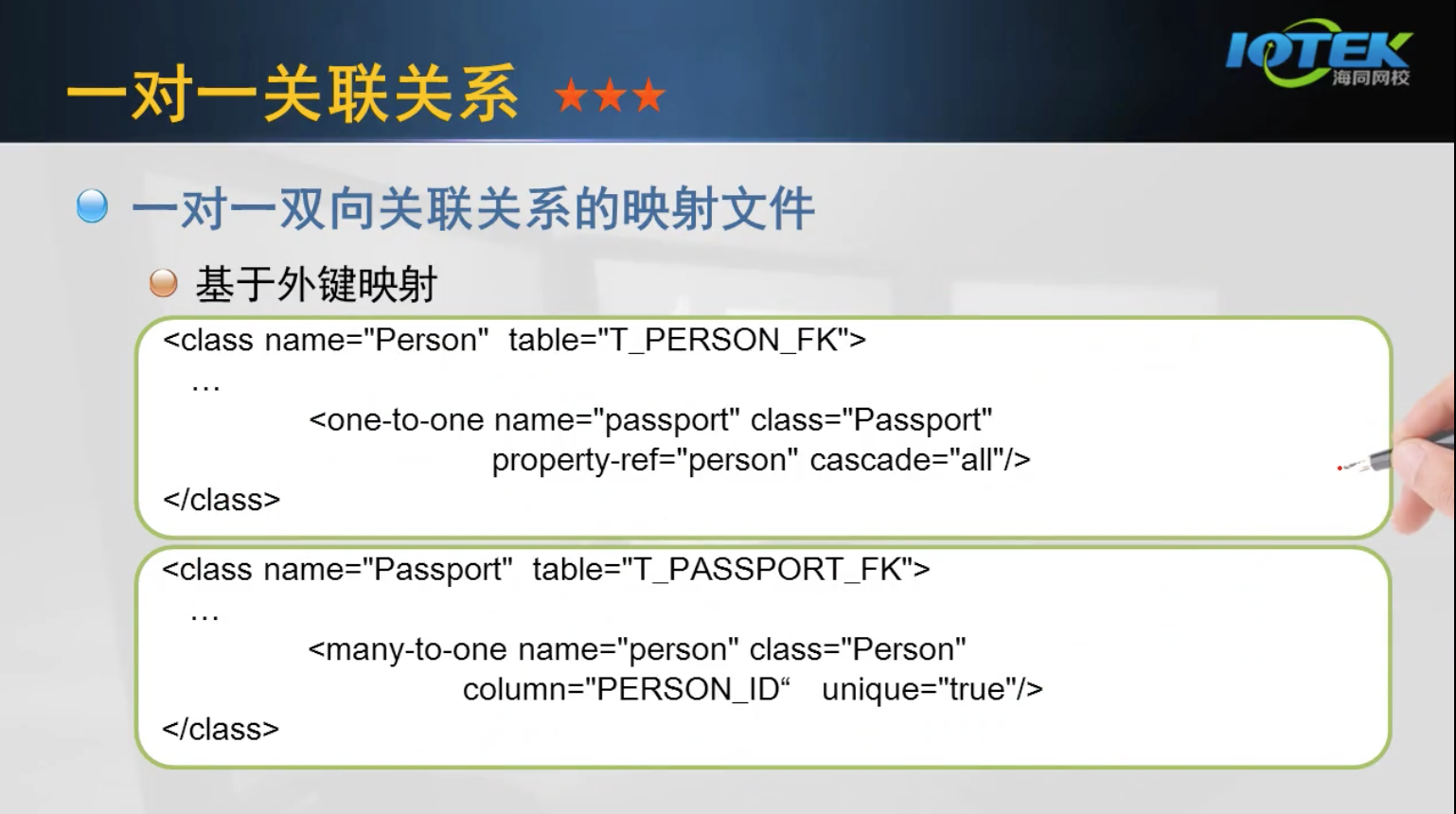


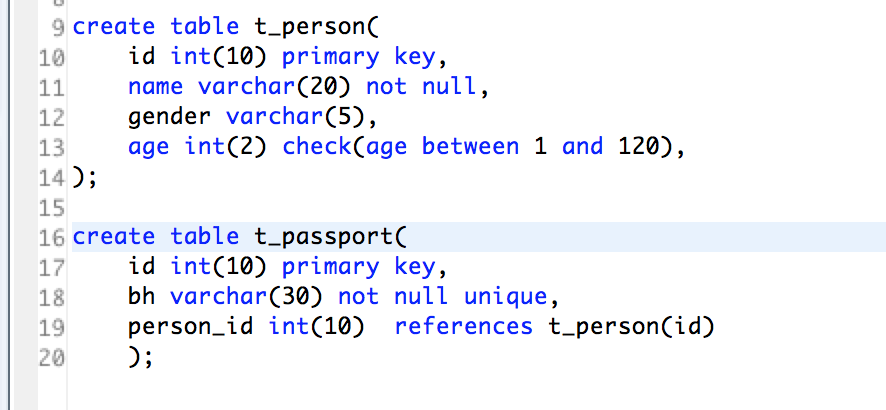
对应的表的创建



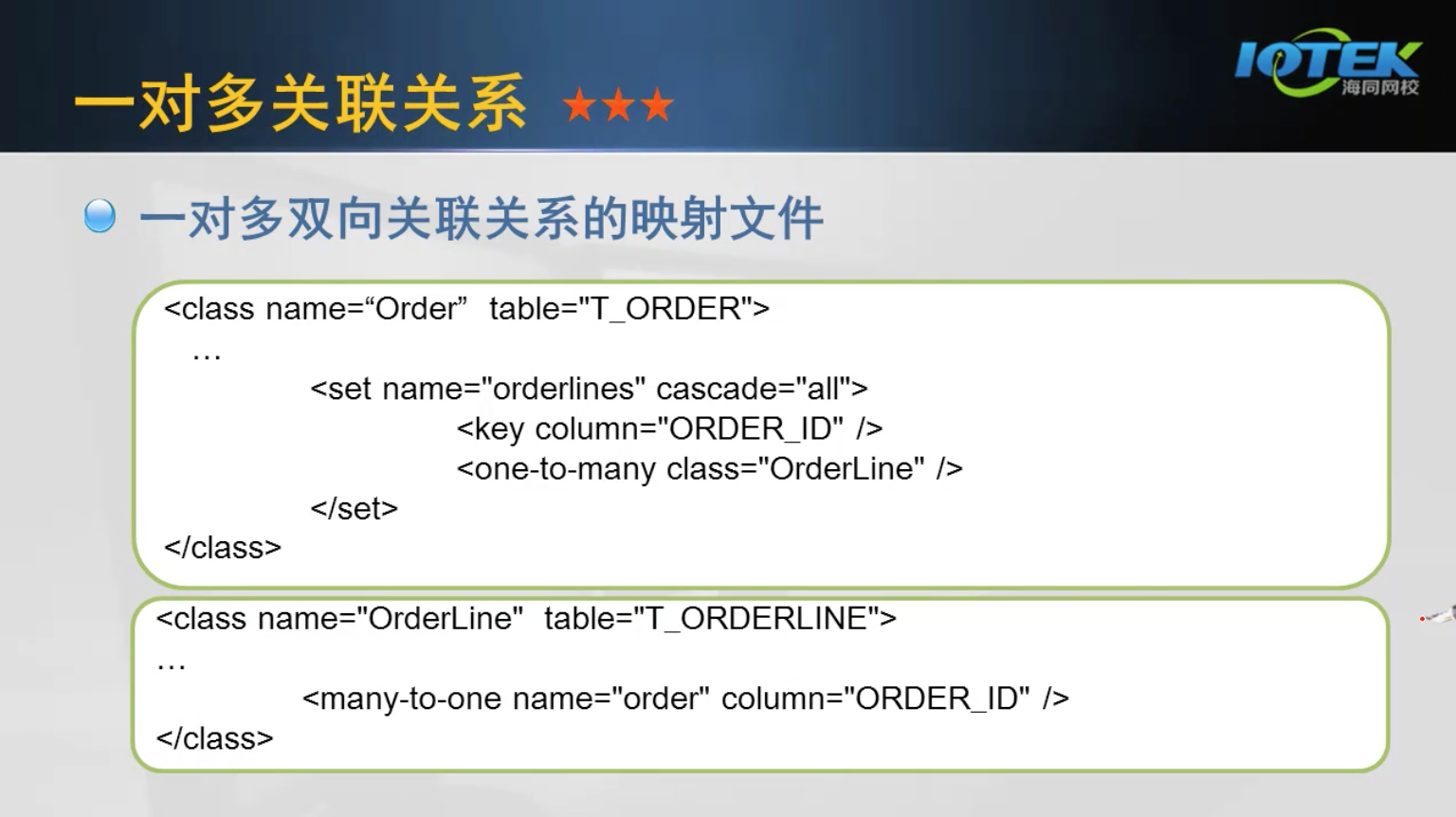
1. 基于外键的映射

就是通常的通过外键链接的方式，在子表中增加一个字段记录父表中的ID作为外键。

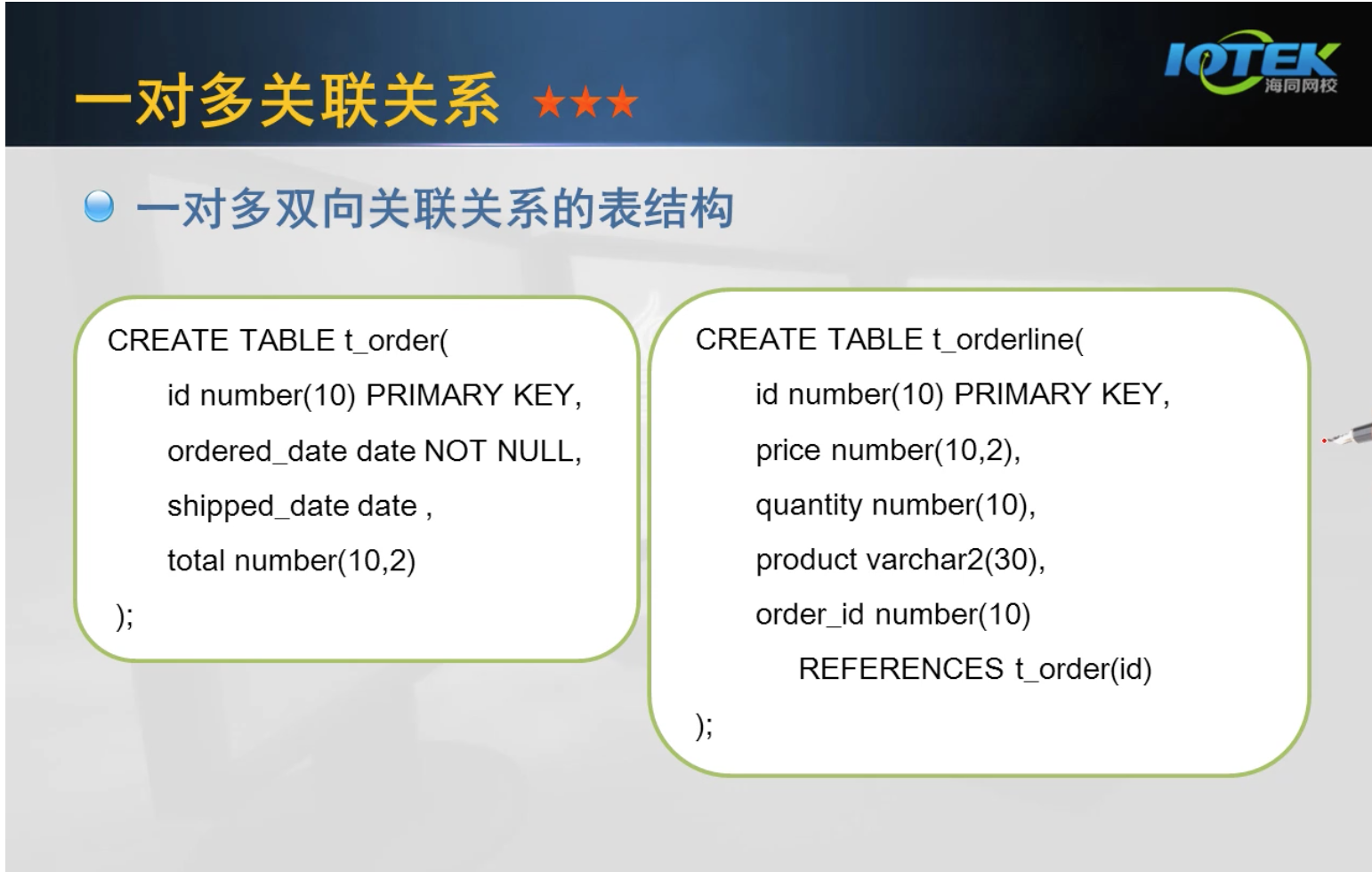


对应表的创建

五、一对多的映射方式



数据库建表

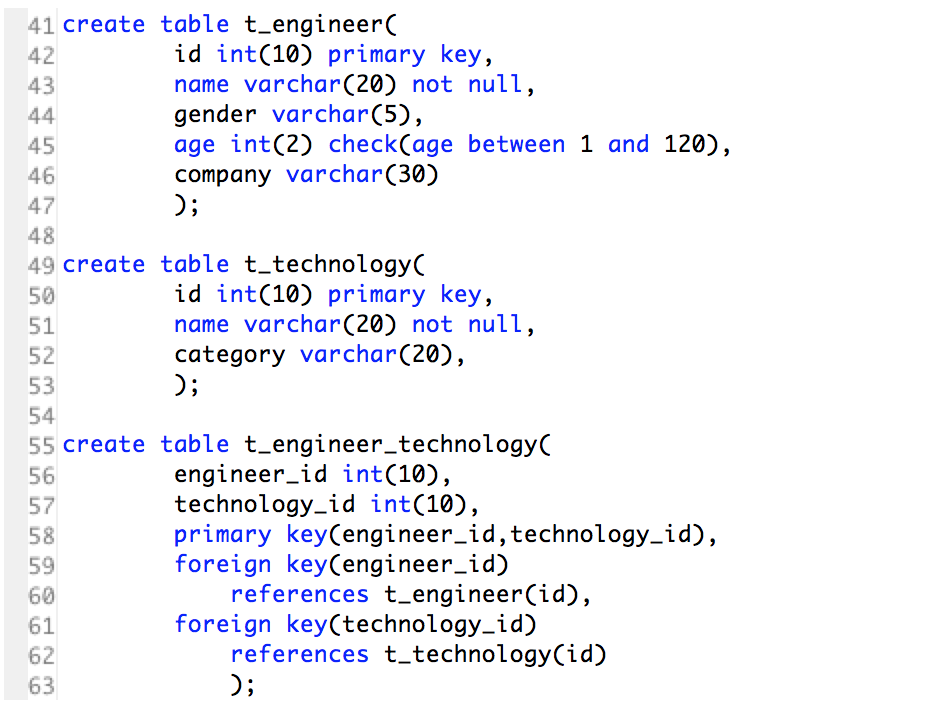


六、多对多的映射

通常，多对多的关系在数据库中有两种方式解决

1. 将其拆分成两个一对多的关系
2. 建立中间的的桥表，进行连接（此处采用这种）

数据表



类结构



映射配置

