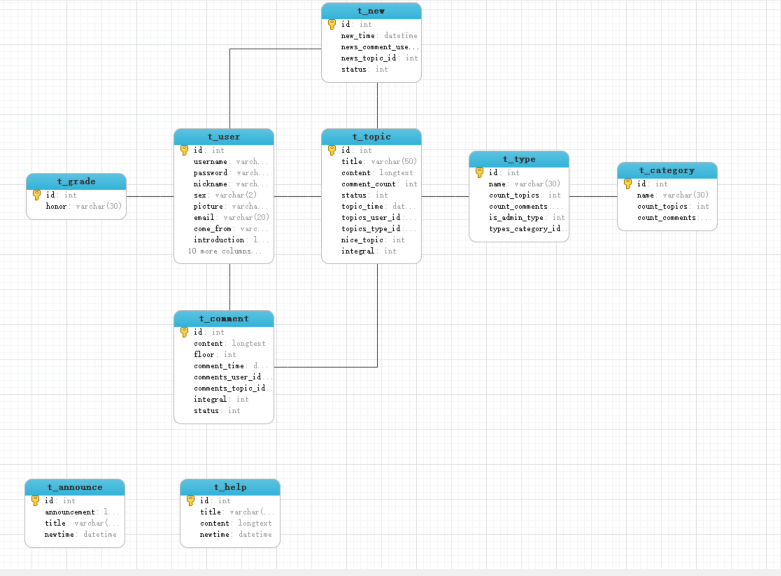
**系统数据访问层——简单工厂模式**

**模式概述**

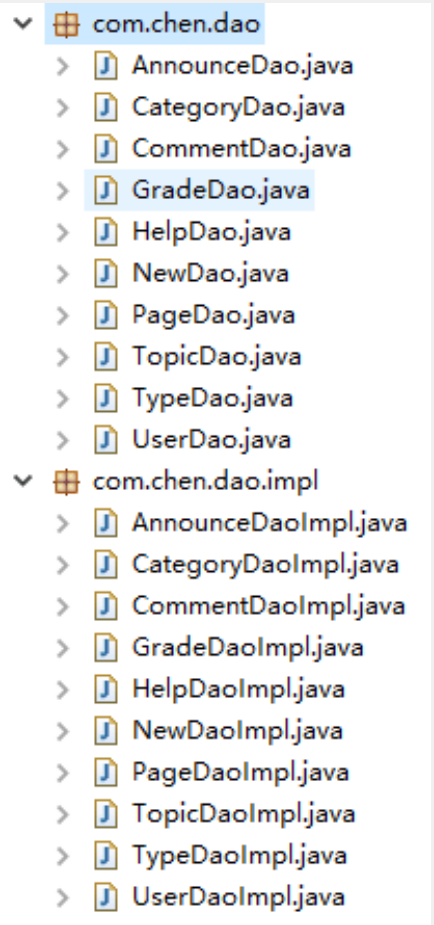
在设计模式中，有一类模式被称为创建型模式，它把对象的创建封装起来，使得创建实例的责任与使用实例的责任分割开来，并由专门的模块负责实例的创建，而系统在宏观上不再依赖于对象创建的细节。工厂模式就属于这种模式，它专门负责将大量有共同接口的类实例化，工厂模式可以动态的决定实例化哪一个类，而不要事先知道每次要实例化的类。工厂模式有3种形态：简单工厂模式、工厂方法、抽象工厂模式。简单工厂模式就是可以根据传入的参数决定创建哪一个类的实例，它其实是工厂方法和抽象工厂模式在一定程度上的简化形式。

**数据访问层的具体设计**

下面介绍简单工厂模式是如何应用于八哥车间BBS论坛系统的数据访问层的。在本系统中，我们把要处理的数据对象分为两类，第一类是数据实体，它对应数据库中相应的数据表，这类对象没有行为，仅仅用来表现对象的数据。这一类的数据对象并不具有持久化的功能，仅仅包含一些属性，也就是说，它们只是作为数据的载体，方便业务逻辑对相应数据表进行读写操作。



第二类数据对象则为数据的业务逻辑对象。这里的业务逻辑，并不是指业务逻辑层上的领域业务逻辑，而是对数据库的基本操作，即Select，Insert，Update和Delete操作。这些对象仅仅包含一些数据库操作而与数据无关，所以它们均被抽象为一个单独的接口模块，放在程序集com.chen.dao中



在该接口中，只包含了一些数据库的增加、删除等基本操作。将数据实体与相关的数据操作分离出来，体现了“职责分离”的原则。将数据实体与其行为分开，使得两者之间依赖减弱，避免了一个类因职责过多、过大，从而导致该类的引用者发生“灾难性”的影响。另外，它体现了“抽象”的思想，即面向接口编程的思想，抽象的接口模块与具体的数据库访问实现完全隔离，保证了系统的可扩展性，同时也保证了数据库的可移植性。

有了数据实体，数据对象的抽象接口和实现，剩下的就只需要对数据对象的创建和管理。

**JAVA DAO模式**

DAO模式。DAO是指位于业务逻辑和持久化数据之间实现对持久化数据的访问，就是将数据库操作都封装起来。

DAO模式提供了访问关系型数据库系统所需操作的接口，将数据访问和业务逻辑分离对上层提供面向对象的数据访问接口，DAO 模式的优势就在于它实现了两次隔离。

1.隔离了数据访问代码和业务逻辑代码。业务逻辑代码直接调用DAO方法即可，完全感觉不到数据库表的存在。分工明确，数据访问层代码变化不影响业务逻辑代码,这符合单一职能原则，降低了藕合性，提高了可复用性。

2.隔离了不同数据库实现。采用面向接口编程，如果底层数据库变化，如由 MySQL 变成 Oracle 只要增加 DAO 接口的新实现类即可，原有 MySQ 实现不用修改。这符合 "开-闭" 原则。该原则降低了代码的藕合性，提高了代码扩展性和系统的可移植性。

一个典型的DAO 模式主要由以下几部分组成。

1.DAO接口： 把对数据库的所有操作定义成抽象方法，可以提供多种实现。

2.DAO 实现类： 针对不同数据库给出DAO接口定义方法的具体实现。

3.实体类：用于存放与传输对象数据。

4.数据库连接和关闭工具类： 避免了数据库连接和关闭代码的重复使用，方便修改。

DAO接口（以公告模块为例）



DAO实现类：



