## 数据库表的构建

数据库的构建和java代码中的实体类的构建有很大的联系。数据库中表的名称和实体类的类名对应，表中属性名和java实体类的属性名对应。

注：java的类名使用的为驼峰命名法，如UserRole转化为数据库中的表名就是user\_role

## Dao层代码书写

BaseDao（包括BaseDao.xml文件）代码书写

介绍：数据库一些基本的增删改查。这个接口是一个公共的接口其他的dao接口都可以继承这个接口。

接口：

1，public void add(@Param("tableName") String tableName, @Param("objects")Object...objects );

向数据库中添加一条数据，objects是数据表对应的value值。

对应的sql语句为：

insert into ${tableName} values(value1,value2,…)

2，public void delete(@Param("tableName") String tableName,@Param("id") Integer id);

删除对应表中的一条记录，id 为该表的主键 索引。

对应的sql语句为：

delete from ${tableName} where id=#{id}

3，public void update(@Param("tableName") String tableName, @Param("id") Integer id, @Param("objects")Object[] objects);

更新对应表中的id对应的那条记录，objects对应 的是set语句集合，如set telephone=’123’

update ${tableName} ${obj1},${obj2}… where id=#{id}

4, public Map<Object, Object> select(@Param("tableName") String tableName, @Param("id") Integer id);

查找对应表中对应id的一条数据，返回键值对。

对应sql语句为

select \* from ${tableName} where id=#{id}

5，public List<Map<Object, Object>> selectAll(@Param("tableName") String tableName);

查找对应表中的所有数据

对应sql语句为

select \* from ${tableName}

6，public void addForNotMatch(@Param("tableName") String tableName,@Param("fieldNames")Object[] fieldNames, @Param("fieldValues")Object[] fieldValues);

向对应表中添加一条记录，指定对应属性的值

insert into ${tableName} () values(value1,value2,…)

## 参考文档地址