在使用JdbcTemplate时，一般传参都是用的？来绑定参数，但是对于某种情况就不适用了，例如Sql中如果存在IN，那么写SQL的时候就会比较麻烦，例如，咱们要查ID在某个范围内的数据，一般情况下咱们这么写：

List<String> ids = new ArrayList<String>();

ids.add("id1");

ids.add("id2");

ids.add("id3");

String sql = "SELECT \* FROM TEST WHERE ID IN(";

for(int i = 0; i<ids.size();i++){

sql+="?";

if(i!=ids.size()-1){

sql+=",";

}

}

sql+=")";

List result = getJdbcTemplate.queryForList(sql, ids.toArray());

这样得自己拼接？，比较麻烦，在Hibernate中，可以使用参数别名的方式传入参数，其实Spring中也可以。

这里需要提到的类就是NamedParameterJdbcTemplate，他是Spring给开发者提供的一个基于JdbcTemplate的类，他支持命名参数特性。包含了JdbcTemplate中的大部分方法，主要有三类：execute方法、query及queryForXXX方法、update及batchUpdate方法。

咱们可以看一个由NamedParameterJdbcTemplate完成的上述例子：

List<String> ids = new ArrayList<String>();</P>

ids.add("id1");

ids.add("id2");

ids.add("id3");

String sql = "SELECT \* FROM TEST WHERE ID IN(:ids)";

Map params = new HashMap();

params.addValue("ids", ids);

NamedParameterJdbcTemplate jdbcTemplate = null;

jdbcTemplate = new NamedParameterJdbcTemplate(getJdbcTemplate());

List reslut = jdbcTemplate.query(sql, params);

可以看到Sql中的参数可以用 :[name] 的方式书写，在执行之前，所有的参数可以放在一个Map中，key为Sql中的参数名，这样的话，就简化了咱们自己拼写Sql的工作量。

其实，这只是NamedParameterJdbcTemplate的好处之一，另一个好处就是，对于同一个参数，多次出现在一条sql中时，可以只设置一次，也很好处理。

例如咱们要查一个数据，传入的时间参数要在字段F1和字段F2之间。那么可以这么写：

String sql = "SELECT \* FROM TEST WHERE F1>:time and F2<:time";

Map params = new HashMap();

params.addValue("time", new Date());

NamedParameterJdbcTemplate jdbcTemplate = null;

jdbcTemplate = new NamedParameterJdbcTemplate(getJdbcTemplate());

List reslut = jdbcTemplate.query(sql, params);

扩展

上面咱们利用NamedParameterJdbcTemplate查询时，是将所有参数放入到了一个Map中，其实NamedParameterJdbcTemplate为咱们提供的参数模型不止Map。还有SqlParameterSource和BeanPropertySqlParameterSource。

其中SqlParameterSource和咱们用Map一样，他只是对Map进行了封装。

而BeanPropertySqlParameterSource封装了一个JavaBean对象，通过JavaBean对象属性来决定命名参数的值。

例如咱们创建了一个Bean。

public class User {

private int id;

private String userName;

private String password;

//省略getter和setter

}

现在验证一个登录信息是否正确：

public boolean login(String userName,String password){

NamedParameterJdbcTemplate namedParameterJdbcTemplate = null;

namedParameterJdbcTemplate = new NamedParameterJdbcTemplate(getJdbcTemplate());

User user = new User();

model.setUserName("jialeens");

model.setPassword("hehe");

String sql = "SELECT COUNT(1) FROM USER WHERE USERNAME=:userName AND PASSWORD=:password";

SqlParameterSource paramSource = new BeanPropertySqlParameterSource(user);

int size = namedParameterJdbcTemplate.queryForInt(sql, paramSource);

return size==1;

}

可以看到，传入的参数是一个对象Bean，Sql在执行时，会根据Sql中的参数名去获取对应Bean中的属性。在有的地方可以使用。