**数据库访问层知识点**

1.\_\_\_\_\_\_\_\_\_用于指定数据库连接池中处于空闲状态的最大连接数目。

2.MyBatis可以通过\_\_\_\_\_\_\_\_来配置和映射原始类型、接口和Java POJO为数据库中的记录。

3.MyBatis是一个基于\_\_\_\_\_\_的持久层框架

4.MyBatis的\_\_\_\_\_\_接口将SqlSession传递的参数动态地生成需要执行的SQL语句。

5.Spring使用声明式事务处理，默认情况下，只有当代码中发生\_\_\_\_\_\_异常，数据库将执行回滚(rollback)操作，撤销以前所有数据库操作

6.MyBatis中用于实现增删改查的基本注解有：\_\_\_\_\_\_，\_\_\_\_\_\_，\_\_\_\_\_\_，\_\_\_\_\_\_。

7.DAO实现，填写数字处的配置。

public **➀** updateUser(User user){

DBUtil dbutil=new DBUtilImpl();

conn = dbutil.getConnection();

boolean flag=false;

if(conn==null)return false;//没有获取连接；

String sql="update tb\_user set password=?, tell=? where userName=?" ;

try{

pst= conn.prepareStatement(sql);

pst.setString(1,user.getPassword());

pst.setString(2,user.getTell());

pst.setString(3,user.getUserName());

pst.executeUpdate();

flag=true;

}catch (SQLException e) {

e.printStackTrace();

} finally {

dbutil.close(conn, pst, rs);

}

return flag;

}}

8.当用Transactional注解方法时，方法体内，不能用try{}catch(){}.解决的办法是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

9.由于业务层是面向对象而数据库是关系，需解决访问接口输入输出的对象-关系的映射问题，引入ORM技术，其基本构成有\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_。

建议答案

1.maxIdle

2. XML或注解

3. Java

4.Executor

5.unchecked

[6.@Insert，@Select，@Update，@Delete](mailto:6.@Insert，@Select，@Update，@Delete)。

7. boolean，

8.让方法体声明抛出异常.让调用者结合其它业务处理

9.数据源、连接池