JDBC 测试题

1. 选择题（30分，请选择每题的所有正确答案）

1、请选出JDBC实现过程中用到的接口或类 b c d

a. System

b. Class

c. Connection

d. ResultSet

2、SELECT COUNT(\*) FROM emp;这条SQL语句执行，如果员工表中没有任何数据， 那么ResultSet中将会是什么样子？c

A.null B.有数据C.不为null，但是没有数据D.以上都选项都不对

3、JDBC的事务实现是在以下哪个接口或类中实现ABC

A. Statement

B. PreparedStatement

C. ResultSet

D. Connection

4、类型为date的字段，对应JDBC的类型为B

A java.sql.Date

B java.util.Date

C 以上都对

D 以上都不对

5、数据库事务的特性包括以下哪些选项 A B C

A 原子性 B 一致性 C 封闭性 D 开放性

6、关于连接池，以下说明正确的是 B D

A.连接池可以提高SQL执行的效率

B. 连接池可以提高JDBC连接的使用效率

C. 为了更好的管理连接池，在连接池管理类中将连接池属性设置为非静态的。

D. 为了限制连接数无限制的增长，在连接池管理类中将连接池属性设置为静态的

二、判断题（10分）

1、预编译声明PreparedStatement的实例不能执行executeQuery(sql)和executeUpdate(sql)方法。X

2、Mysql事务默认为自动提交，每执行一条更新SQL语句就会启动一个事务，执行完毕后自动提交，如果出现异常则自动回滚。 X

3、Connection关闭后，ResultSet可以继续使用。 O

4、JDBC程序只能连接Mysql数据库，不能连接其他类型的数据库。 X

5、JDBC封装中PreparedStatement采用setObject方法设置参数值，是因为Object可以为不同类型的参数赋值。 O

三、简答题（60分）

1. 简要说明连接池管理类中关闭连接后为什么要从连接池中删除。

为了限制连接数无限制的增长，在连接池管理类中将连接池属性设置为静态的，所以连接池管理类中关闭连接后需要删除链接避免占用空间。

1. 请简要说明DAO接口和实现类在JDBC中的功能，并说明使用DAO如何实现对相关数据的操作。

答：DAO主要实现对数据库的增删改查，往数据库对应的类中添加数据。

1. 请简要说明Connection、Statement、ResultSet关闭的思路，并用代码说明。

ResultSet不为null时优先关闭、释放，

if（ResultSet ！= null）{

ResultSet.Close（）；

ResultSet = null ；

Statement不为null时关闭、释放，

if（Statement ！= null）{

Statement.Close（）；

Statement = null ；

Connection不为null时关闭、释放，

if（Connection ！= null）{

Connection.Close（）；

Connection = null ；

1. 说明使用连接池管理连接的实现思路。
2. 为了限制连接数无限制的增长，在连接池管理类中将连接池属性设置为静态的
3. 使用过后链接池会删除不使用的链接。
4. 请简要说明事务的特性，mysql中事务是如何实现？JDBC如何实现事务？

事务的特性：原子性 一致性 封闭性 持久性。

Mysql事务：（1）设置autoCommit变量的值为OFF；（2）事务通过commit提交，通过rollback回滚。

JDBC：事务在同一个Connection对象实现的（1）Connection.setAutoCommit(false);(2)通过Connection的commit()提交，rollback()方法回滚。

1. 写一个连接Mysql数据库在UserInfo表中完成注册、完善 用户信息的Jdbc程序

Connection con = **null**;

PreparedStatement pstmt = **null**;

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

**try** {

con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/a3");

String sql = "insert into userInfo (userName , userPass) value (?,?)";

pstmt = con.prepareStatement(sql);

pstmt.setString(1,"xiaoming");

pstmt.setString(2,"oass");

**int** i = pstmt.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

**try** {

pstmt.close();

con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}