

数据库应用程序设计

一. 5*6=30

1. 加载用于application的DB2程序

```
static
{ try
{ Class.forName ("COM.ibm.db2.jdbc.app.DB2Driver");
}
catch (Exception e)
{ System.out.println ("\n Error loading DB2 Driver...\n");
System.out.println (e);
System.exit(1); }
}
```

2. 无参数标记的SQL语句，应调用哪个类中什么方法实例化一个Statement对象，对象名自行定义

```
Connection con =
DriverManager.getConnection("jdbc:db2:sample","db2admin","db2a
dmin");
Statement stmt = con.createStatement();
```

3. 定义一个结果集对象，该对象查询语句是“查询表staff，获取员工编号为10 的名字 (name)，工作 (job)，工资 (salary) ”

```
ResultSet rs = stmt.executeQuery("select NAME, JOB, SALARY
from administrator.staff where empno = 10");
```

4. 将程序设置为不自动提交SQL语句

```
con.setAutoCommit(false);
```

5. 假设Connection，Statement，ResultSet类对象分别为，con，stmt，rs，应该按什么顺序关闭对象，写出具体语句

```
rs.close();
```

```
stmt.close();
```

```
con.close();
```

6. 写出读取结果集下一行的语句，如何判空

```
rs.next();
while(rs.next()){
.....
}
rs.isNull();
```

注：结果集

first() 移到第一行

last() 移到最后一行

previous(): 移动到前一行

beforeFirst(): 移动到第一行之前，常用于从头开始读取行

afterLast():移动到最后一行之后

absolute(int rowNum):移动到行数为rowNum的行；如果rowNum为负值，则以最后一行为首、第一行为尾

relative(int relativeRowNum):相对结果集当前所在行，移动行数为relativeRowNum的行

二. $8 \times 5 = 40$

1.编写程序片段，完成修改操作“根据用户输入的部分编号（dept）将表staff中的工资（salary）上涨5%”，最后输出被成功修改的行数，假设变量mydeptno用来存放用户输入的部门编号

```
String deptno = ""; // 用来存放输入的部门编号
String s = " ";
int mydeptno = 0; //用来存放部门编号的变量
String sqlstmt = "UPDATE ADMINISTRATOR.STAFF SET SALARY = SALARY * 1.05 WHERE DEPT = ?";
BufferedReader in = new BufferedReader( new InputStreamReader
(System.in));
Connection sample =
DriverManager.getConnection("jdbc:db2:sample","db2admin","db2a
dmin");
System.out.println("input deparment no");
s = in.readLine();
int updateCount=0; //用来统计修改的行数
While(s!=null){
deptno = s.substring(0,2);
mydeptno = Integer.parseInt(deptno);
PreparedStatement pstmt = sample.prepareStatement( sqlstmt );
pstmt.setInt(1, mydeptno);
updateCount += pstmt.executeUpdate();
System.out.println("input deparment no");
s = in.readLine();
}
System.out.println("\nNumber of rows updated: " +
updateCount);
```

2.编写程序片段，针对1题，当输入的部门编号有误时，以GUI的形式提示“您所输入的部门编号有误”需要考虑异常的处理

```
try{
PreparedStatement pstmt =
sample.prepareStatement( sqlstmt );
pstmt.setInt(1, mydeptno);
updateCount += pstmt.executeUpdate();
System.out.println("\nNumber of rows updated: " +
updateCount);
JOptionPane.showMessageDialog(null, "共修改"+ updateCount,
"54130409", JOptionPane.INFORMATION_MESSAGE);
}catch( SQLException x ){
SQLCode = x.getErrorCode();
SQLState = x.getSQLState();
```

```

        String Message = x.getMessage();
        System.out.println("\nSQLCODE: " + SQLCode );
        System.out.println("\nSQLSTATE: " + SQLState);
        System.out.println("\nSQLERRM: " + Message);
        JOptionPane.showMessageDialog(null, "输入部门编号有误",
"54130409", JOptionPane.INFORMATION_MESSAGE);
    }

```

3.有如下的DDL:

create table templ (empno char (6) , firstname varchar (20) ,
lastname varchar (20) , salary decimal (7, 2) birthday date) ;

现有一新员工Roth（可看作是lastname的值）等待插入表TEMP，其中员工编号为000110，工资是50000，请编写该程序片段

```

String sql = "INESRT INTO TEMPL (lastname ,empno ,SALARY)
VALUES (ROTH ,000110,50000)";
Statement stmt = Connection.createStatement();
Int i = stmt.executeUpdate(sql);

```

4.先对表EMP进行查询，获取包含LASTNAME，FIRSTNAME两列的结果集，然后只针对LASTNAME是“SMITH”的行进行修改，将该行的FIRSTNAME改为用户输入的字符串；

```

String mySelect = "SELECT LASTNAME, FIRSTNME FROM EMP";
String myUpdate = "UPDATE EMP SET FIRSTNME = ? WHERE CURRENT
OF ";
String cursorName = null;
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(mySelect);
cursorName = rs.getCursorName();
PreparedStatement ps = con.prepareStatement(myUpdate +
cursorName);
While (rs.next()) {
String lastname = rs.getString(1);
String firstnme = rs.getString(2);
if (lastname.equals("SMITH")) {
String newFirstnme = "George";
ps.setString( 1, newFirstnme);
ps.executeUpdate();
}
}
rs.close();
ps.close();
stmt.close();

```

5.编写程序片段，完成查询语句

“SECLET EMPNO, LASTNAME, EDLEVEL, FROM EMP”，并将查询结果进行输

出，当EDLEVEL是否为空值时，提示给用户相关信息，例如如果职工编号为0100的员工没有Edlevel值，则显示：“Edlevel is null forEMPNO 0100”.

```
ResultSet rs =  
stmt.executeQuery("SECTET EMPNO, LASTNAME, EDLEVEL, FROM EMP");  
  
While(rs.next()){  
  
String empno = rs.getString(1);  
  
String lastname = rs.getString(2);  
  
String edlevel = rs.getString(3);  
  
If(edlevel==null){  
  
"输入为空";}  
  
System.out.println(empno+lastname+edlevel);  
  
}
```

三、 $15 \times 2 = 30$

如图所示sample数据库的部分表结构以及他们之间的参照关系，其中表格中标有下划线的是主键，倾斜字体的为外键。编写完整的程序代码，实现如下操作：

(1) 新员工的插入操作，其中：

新员工的入职照片所在路径为“e:\empphoto”，照片的类型可以是jaeg, gif, bmp三种类型；

当照片的大小超出限制时，要提示给用户；

员工的其他信息应该由用户通过键盘输入、

(2) 员工的查询功能：根据用户输入的部门名称和工龄，显示出该部门中大于此工龄的员工姓名、工龄和电话，并按照工龄升序排列。

