

第一个存储过程

```
1  --第一个存储过程：打印Hello World
2  /*
3  调用存储过程：
4  1. exec sayhelloworld();
5  2. begin
6      sayhelloworld();
7      sayhelloworld();
8  end;
9  /
10 */
11 create or replace procedure sayhelloworld
12 as
13 --说明部分
14 begin
15     dbms_output.put_line('Hello World');
16 end;
17 /
18
```

带参数的存储过程

```
1  --创建一个带参数的存储过程：
2  --给指定的员工涨100块钱的工资，并且打印涨前和涨后的薪水
3  create or replace procedure raisesalary(eno in number)
4  as
5  --定义一个变量保存涨前的薪水
6  psal emp.sal%type;
7  begin
8  --得到员工涨前的薪水
9  select sal into psal from emp where empno=eno;
10
11 --给该员工涨100
12 update emp set sal=sal+100 where empno=eno;
13
14 --需不需要commit?
15 --注意：一般不在存储过程或者存储函数中，commit和rollback。
16
17 --打印
18 dbms_output.put_line('涨前：'||psal||' 涨后：'||(psal + 100));
19
20 end;
21 /
```

存储函数

```
1 create or replace function queryempincome(eno in number)
2 return number
3 as
4 --定义变量保存员工的薪水和奖金
5 psal emp.sal%type;
6 pcomm emp.comm%type;
7 begin
8 --得到该员工的月薪和奖金
9 select sal,comm into psal,pcomm from emp where empno=eno;
10
11 --直接返回年收入
12 return psal*12+pcomm;
13 end;
```

注意：return 带有null的表达式，结果为null。用nvl(param,0)函数，把null当做0处理。

输出参数

```
1 create or replace procedure queryempinform(eno in number,
2                                           pename out varchar2,
3                                           psal out number,
4                                           pjob out varchar2)
5 as
6 begin
7 --得到该员工的姓名 月薪和职位
8 select ename,sal,empjob into pename,psal,pjob from emp where empno=eno;
9 end;
```

在java中访问存储过程

```

1 package demo.utils;
2
3 import java.sql.DriverManager;
4
5 public class JDBCUtils {
6     private static String driver = "oracle.jdbc.OracleDriver";
7     private static String url = "jdbc:oracle:thin:@192.168.56.101:1521:orcl";
8     private static String user = "scott";
9     private static String password = "tiger";
10
11     //注册数据库的驱动
12     static{
13         try {
14             Class.forName(driver);
15             //DriverManager.registerDriver(driver);
16         } catch (ClassNotFoundException e) {
17             throw new ExceptionInInitializerError(e);
18         }
19     }
20
21
22     //获取数据库连接
23     public static Connection getConnection(){
24         try {
25             return DriverManager.getConnection(url, user, password);
26         } catch (SQLException e) {
27             e.printStackTrace();
28         }
29         return null;
30     }
31
32
33     //释放数据库的资源
34     public static void release(Connection conn,Statement st,ResultSet rs){
35         if(rs != null){
36             try {
37                 rs.close();
38             } catch (SQLException e) {
39                 e.printStackTrace();
40             }finally{
41                 rs = null;
42             }
43         }
44         if(st != null){
45             try {
46                 st.close();
47             } catch (SQLException e) {
48                 e.printStackTrace();
49             }finally{
50                 st = null;
51             }
52         }
53         if(conn != null){
54             try {
55                 conn.close();
56             } catch (SQLException e) {
57                 e.printStackTrace();
58             }finally{
59                 conn = null;
60             }
61         }
62     }
63 }
64

```

访问存储过程

```

1 package demo.oracle;
2
3 import org.junit.Test;
4
5 public class TestProcedure {
6     /*
7      * create or replace procedure queryempinform(eno in number,
8                                             pename out varchar2,
9                                             psal      out number,
10                                            pjob      out varchar2)
11     */
12     @Test
13     public void testProcedure(){
14         //{call <procedure-name>[(<arg1>,<arg2>, ...)]}
15
16         String sql = "{call queryempinform(?,?,?,?)}";
17
18         Connection conn = null;
19         CallableStatement call = null;
20         try {
21             //得到一个连接
22             conn = JDBCUtils.getConnection();
23             //通过连接创建出statement
24             call = conn.prepareCall(sql);
25
26             //对于in参数, 赋值
27             call.setInt(1, 7839);
28
29             //对于out参数, 申明
30             call.registerOutParameter(2, OracleTypes.VARCHAR);
31             call.registerOutParameter(3, OracleTypes.NUMBER);
32             call.registerOutParameter(4, OracleTypes.VARCHAR);
33
34             //执行调用
35             call.execute();
36
37             //取出结果
38             String name = call.getString(2);
39             double sal = call.getDouble(3);
40             String job = call.getString(4);
41             System.out.println(name+"\t"+sal+"\t"+job);
42         } catch (Exception e) {
43             // TODO: handle exception
44             e.printStackTrace();
45         } finally{
46             JDBCUtils.release(conn, call, null);
47         }
48     }
49 }
50
51 }
52
53 }
54
55 }

```

访问存储函数

```

1 package demo.oracle;
2
3 import java.sql.CallableStatement;
4 import java.sql.Connection;
5
6 import org.junit.Test;
7
8 public class TestFunction {
9     /*
10      * create or replace function queryempincome(eno in number)
11      return number
12      */
13     @Test
14     public void testFunction(){
15         //{?= call <procedure-name>[(<arg1>,<arg2>, ...)]}
16
17         try {
18             //得到数据库连接
19             conn = JDBCUtils.getConnection();
20             //基于连接创建statement
21             call = conn.prepareCall(sql);
22
23             //对于输出参数，申明
24             call.registerOutParameter(1, OracleTypes.NUMBER);
25             //对于输入参数，赋值
26             call.setInt(2, 7839);|
27             //执行调用
28             call.execute();
29             //取出年收入的结果
30             double income = call.getDouble(1);
31             System.out.println("该员工的年收入是:"+income);
32         } catch (Exception e) {
33             e.printStackTrace();
34         } finally{
35             JDBCUtils.release(conn, call, null);
36         }
37     }
38 }
39
40
41
42
43
44 }

```

在out参数中使用光标

- 申明包结构
- 包头
- 包体

- 包头：

```
CREATE OR REPLACE PACKAGE MYPACKAGE AS

    type empcursor is ref cursor;
    procedure queryEmpList(dno in number, empList out empcursor);

END MYPACKAGE;
```

- 包体：

```
CREATE OR REPLACE PACKAGE BODY MYPACKAGE AS

    procedure queryEmpList(dno in number, empList out empcursor) AS
    BEGIN
        open empList for select * from emp where deptno=dno;
    END queryEmpList;

END MYPACKAGE;
```

```
1 CREATE OR REPLACE
2 PACKAGE MYPACKAGE AS
3
4     type empcursor is ref cursor;
5     procedure queryEmpList(dno in number, empList out empcursor);
6
7 END MYPACKAGE;
```

```
1 CREATE OR REPLACE
2 PACKAGE BODY MYPACKAGE AS
3
4     procedure queryEmpList(dno in number, empList out empcursor) AS
5     BEGIN
6
7         --打开光标
8         open empList for select * from emp where deptno=dno;
9
10    END queryEmpList;
11
12 END MYPACKAGE;
```

在java中访问包中的存储过程

```

1 package demo.oracle;
2
3 import org.junit.Test;
4
5 public class TestCursor {
6     /*
7     CREATE OR REPLACE PACKAGE MYPACKAGE AS
8
9     type empcursor is ref cursor;
10    procedure queryEmpList(dno in number, empList out empcursor);
11
12    END MYPACKAGE;
13    */
14    @Test
15    public void testCursor(){
16
17        // {call <procedure-name>[(<arg1>,<arg2>, ...)]}
18        String sql = "{call MYPACKAGE.queryEmpList(?,?)}";
19
20        Connection conn = null;
21        CallableStatement call = null;
22        ResultSet rs = null;
23        try {
24            // 获取数据库的连接
25            conn = JDBCUtils.getConnection();
26            // 创建statement
27            call = conn.prepareCall(sql);
28
29            // 对于in参数, 赋值
30            call.setInt(1, 10);
31            // 对于out参数, 申明
32            call.registerOutParameter(2, OracleTypes.CURSOR);
33            // 执行调用
34            call.execute();
35
36            // 取出该部门中所有员工的信息
37            rs = ((OracleCallableStatement)call).getCursor(2);
38            while(rs.next()){
39                // 取出该员工的员工号 姓名 薪水和职位
40                int empno= rs.getInt("empno");
41                String name = rs.getString("ename");
42                double salary=rs.getDouble("sal");
43                String job = rs.getString("empjob");
44                System.out.println(empno+"\t"+name+"\t"+salary+"\t"+job);
45            }
46        } catch (Exception e) {
47            e.printStackTrace();
48        } finally{
49            JDBCUtils.release(conn, call, rs);
50        }
51    }
52 }

```

注意：记得带上包名。