存储过程入门

例 1.--创建一个返回结果集的存储过程 prSearchcontentshow

```
create procedure prSearchcontentshow
AS
begin
        select * from contentshow
end
```

--执行 prSearchcontentshow
execute prSearchcontentshow

例 2.--创建一个要求输入一个输入参数的存储过程 seachConShow

```
create procedure seachConShow
  @divid varchar(10) --定义一个输入参数
as
     select * from contentshow
       where id = @divid --要求 ID 列与输入参数相等
```

--执行 seachConShow 存储过程
execute seachconshow '1'

例 3.--创建一个要求输入两个输入参数的存储过程 searchConShow

```
create procedure searchConShow
  @divid varchar(10), --定义一个输入参数
  @divname varchar(10)--定义的另一个输入参数
as
     select * from contentshow
       where id = @divid and divname =@divname--要求 ID 列与输入参数相等
```

--运行带参数的存储过程
execute searchConShow '1','divid'

execute searchconshow '2','divid1'

存储过程入门

例 4.--创建有返回值的存储过程

```
create procedure getid
        @divid varchar(10),
        @divname varchar(10),
        @topmenu int output
as
        select @topmenu=topmenuid from contentshow
                where id = @divid and
                        divname = @divname
end
```

--执行 getid 这个带返回值的存储过程
Declare @topmenu int --声明一个变量用来接收执行过存储过程后的返回值
execute getid '1','divid',@topmenu output
select @topmenu as 'topmenuid'--as 后是给返回的列值起的一个别名

例 4.1.--修改已经创建过的存储过程

```
alter procedure getid
        @divid varchar(10),
        @divname varchar(10),
        @topmenu varchar output
as
        select @topmenu=topmenuid from contentshow
                where id = @divid and
                        divname = @divname
```

例 4.2.--修改已经创建过的存储过程

```
alter procedure getid
        @divid varchar(10),
        @divname varchar(10),
        @topmenu int output
as
        select @topmenu=topmenuid from contentshow
                where id = @divid and
                        divname = @divname
```

select * from contentshow

存储过程入门

```
create procedure getdivname
    --定义两个输入参数
     @divid    varchar(10),
     @divname varchar(10)
AS
     declare @returndivname varchar--定义一个 varchar 类型的变量

     select @returndivname = divname
     from contentshow
     where
     id = @divid and divname = @divname

return @returndivname
```

```
alter procedure getdivname
    --定义两个输入参数
     @divid    varchar(10),
     @divname varchar(10)
AS
     declare @returndivname int--定义一个 int 类型的变量

     select @returndivname = id
     from contentshow
     where
     id = @divid and divname = @divname
--return 语句可以接收一个整形表达式(int ,smallint,tinyint),而不是一个整形值
return @returndivname

end
```

```
--执行 getdivname 存储过程
declare @id int
execute @id = getdivname '1','divid'
select @id as id
```

--存储过程输入参数添加默认值

存储过程入门

```
create procedure prGetUsers
	@id varchar(10) = '%',
	@username varchar(10) = '%'
AS
	Select * from users
	where id = @id    and
		Username = @username


--修改 prGetUsers 存储过程
alter procedure prGetUsers
	@id int = 2,
	@username varchar(10) = '%'
AS
	Select * from users
	where id = @id    and
		Username = @username


execute prGetUsers    2,'admin'


execute prGetUsers    1,'admin'




--创建
create procedure prGetUser
	@id int = '%',
	@username varchar(10) = '%'
AS
	Select * from users
	where id like @id    and
		Username like @username


--执行
execute prGetUser 2,'a%'



--修改
```

存储过程入门

```
alter procedure prGetUser
	@id int = 2,
	@username varchar(10) = '%'
AS
	Select * from users
	where id like @id    and
		Username like @username


execute prGetUser


select * from users where username like '%a%'

select * from users where username like 'a%'
```

```
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.ResultSet;
import java.sql.Connection;
//调用存储过程要调用的包
import java.sql.CallableStatement;
import java.sql.Types;

import cn.com.xiangxinli.www.db.Db_DB;
import cn.com.xiangxinli.www.db.Db_JDBC;

/**
 * 用JAVA调用存储过程代码
 *
 * @author Administrator
 *
 */
public class procedure {

    /**
     * 得到数据库连接
     * @return Connection
     */
    public Connection getConn()
    {   Connection conn = null;
```

```
        Db_JDBC jdbc = new Db_JDBC();
        try {
            Class.forName(jdbc.sDBDriver);
            try {
                conn =
DriverManager.getConnection(jdbc.connStr,jdbc.username,jdbc.password)
;
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            //System.out.print("OK");
            }
            catch(java.lang.ClassNotFoundException e) {
            System.err.println("Db_DB(): " + e.getMessage());
            }
        return conn;
    }

    /**
     * --创建一个返回结果集的存储过程prSearchcontentshow
        create procedure prSearchcontentshow

        AS

        begin
             select * from contentshow
        end

        --执行prSearchcontentshow
        execute prSearchcontentshow
     *
     *
     * 调用返回一个结果集存储过程prSearchcontentshow
     * @param conn
     */
    public void getResultSet(Connection conn) {
        try {
            Db_DB db = new Db_DB();
            String sql = "{ call prSearchcontentshow }";
            CallableStatement st = conn.prepareCall(sql);
            ResultSet rs = st.executeQuery(sql);
            if (rs.next()) {
                do
```

```
                {
                System.out.println(rs.getString("id") + ";"
                        + rs.getString("divname") + ";"
                        + rs.getString("topmenuid") + ";"
                        + rs.getString("topmenuid1"));
                }while(rs.next());
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    /**
     * --创建一个要求输入一个输入参数的存储过程seachConShow
        create procedure seachConShow

         @divid varchar(10) --定义一个输入参数
        as
           select * from contentshow
            where id = @divid --要求ID列与输入参数相等

        --执行seachConShow存储过程
        execute seachconshow '1'
     *
     * 调用提供一个输入参数的存储过程
     */
    public void oneInputParameter(String para,Connection conn) {
        try {
            Db_DB db = new Db_DB();
            //调用存储过程sql语句写法
            String sql = "{call seachConShow(?)}";
            //声明CallableStatement对象
            CallableStatement cas = conn.prepareCall(sql);
            //给存储过程传入需要的参数
            cas.setString(1, para);
            ResultSet rs = cas.executeQuery();
            if (rs.next())
            {
                do
                {

    System.out.println(rs.getString("id")+";"+rs.getString("divname")
);
```

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

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    /**
     * --创建一个要求输入两个输入参数的存储过程searchConShow
       create procedure searchConShow

        @divid varchar(10), --定义一个输入参数
        @divname varchar(10)--定义的另一个输入参数
       as
          select * from contentshow
           where id = @divid and divname =@divname--要求ID列与输入参数相
等



       --运行带参数的存储过程
       execute searchConShow '1','divid'

       execute searchconshow '2','divid1'
     *
     *
     * 调用提供二个输入参数的存储过程
     */
    public void twoInputParameter(String para,String para1,Connection
conn) {
        try {
            Db_DB db = new Db_DB();
            //调用存储过程sql语句写法
            String sql = "{call searchConShow(?,?)}";
            //声明CallableStatement对象
            CallableStatement cas = conn.prepareCall(sql);
            //给存储过程传入需要的参数
            cas.setString(1, para);
            cas.setString(2,para1);
            ResultSet rs = cas.executeQuery();
            if (rs.next())
            {
```

```
                do
                {

    System.out.println(rs.getString("id")+";"+rs.getString("divname")
);
                }while(rs.next());
            }

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    /**
     *
     * --创建有返回值的存储过程
        create procedure getid
            @divid varchar(10),
            @divname varchar(10),
            @topmenu int output
        as
            select @topmenu=topmenuid from contentshow
                where id = @divid and
                      divname = @divname

        --执行getid这个带返回值的存储过程
        Declare @topmenu int --声明一个变量用来接收执行过存储过程后的返回值
        execute getid '1','divid',@topmenu output
        select @topmenu as 'topmenuid'--as 后是给返回的列值起的一个别名
     *
     *
     * 调用提供二个输入参数的存储过程
     */
    public void returnOneOutputParameter(String para,String
para1,Connection conn) {
        try {
            Db_DB db = new Db_DB();
            //调用存储过程sql语句写法
            String sql = "{call getid(?,?,?)}";
            //声明CallableStatement对象
            CallableStatement cas = conn.prepareCall(sql);
            //给存储过程传入需要的参数
            cas.setString(1, para);
```

```
            cas.setString(2,para1);
            cas.registerOutParameter(3, Types.INTEGER);
            cas.execute();
            System.out.println("返回的值是："+cas.getInt(3));

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        procedure proce = new procedure();
        Connection conn = proce.getConn();
        //proce.getResultSet(conn);
        //proce.oneInputParameter("1", conn);
        //proce.twoInputParameter("2", "divid1", conn);
        proce.returnOneOutputParameter("3", "divid2", conn);
    }

}
```