Oracle的数据操作

1. Oracle的PL/SQL编程
2. PL/SQL的介绍
3. PL/SQL叫做过程化SQL语言(Procedural Language/SQL)，他解决了纯的sql的技术缺陷（针对来说1.不能模块化编程2.执行效率不高3.安全性不高4.浪费带宽）
4. PL/SQL的变量分四种，2 3 4放在包里就可以跨存储过程
5. 标量《就是oracle的sql三大类》
6. 复合类型
7. 记录《像其他语言的类（对象更加合适）》语法如下：

type 记录名 is record(

v\_ename emp.ename %type;

v\_sal emp.sal %type;

… …

)

《v\_emp定义成记录类型：v\_emp 记录名 🡺 加入 🡺 取出：记录名.在记录里你要的那个标量》

1. 表《像数组少用》语法如下：

type 表名 is table of emp.ename %type index by binary\_integer;

《定义：v\_arr b表名🡺加入v\_arr=10🡺取出v\_arr(-1);

1. 参照类型
2. 游标 语法如下：

type 游标名 is ref cursor;《1定义》

open 游标变量名 for select语句《2打开》

fetch 游标变量名 into 其他变量《3一行一行的值转给其他变量》

游标变量名 % notfound;《4如果游标已经指向最后一条的后一条，则返回真》

close 游标变量名；《关闭游标变量》

1. lob

* 技巧：使用《%temp》;

1. PL/SQL的控制语句
2. If…then/if…then…else/if…then…elsif
3. loop循环：

loop

执行

再判断：exit when 条件表达式

end loop;

1. while循环：

while

条件表达式 loop

执行

自加

end loop;

1. for循环：不推荐

for I in reverse 1..10 loop

执行

end loop;

1. case 语句不学了，哈哈我太东那个语法
2. 块的介绍
3. 块是（过程，函数，触发器，包）的组成单位
4. 基本语法：

declare

/\*定义 变量，常量，游标，例外…\*/

begin

--执行部分

exception

--例外处理部分

end;

1. 存储过程的介绍
2. 本语法：

create procejure N(输入参数/输出参数 n个…) is

/\*创建，定义 变量，常量，游标，例外…\*/

begin

--执行部分

exception

--例外处理部分

end;

1. 控制台调用：exce N(实参)/call(实参)
2. java调用：

sql=“{call N(？,？…)}”

创建sql语句对象：cs=ct.propareCall(sql);

给输入问号赋值：sc.setString(1,值);《输入的参数，统一这样来赋值》

给输出的问号赋值注册：cs.regiterOutParameter(2,值)；《给输出参数注册值》

Boolean i=cs.execute();/ResultSet rs=cs.exceute();

1. 函数的介绍
2. 本语法：

create function N(输入参数 n个…输出参数不在这里)

return 输出的参数类型 is

/\*创建，定义 变量，常量，游标，例外…\*/

begin

--执行部分;

return 要输出的值；

end;

1. 控制台调用：select N(参数) from dual；
2. java调用：

sql=”select N(参数) from dual”;

ps=ct.prepareStatement(sql);

ps.setString(1,值);《循环给参数统一赋值输入输出都一样赋值》

i=execute();/rs=execute();

1. 包java调用方法ps(输入输出都赋值)/存储过程cs(输入赋值/输出注册)
2. 本语法：

create package N is

--声明函数

--声明存储过程

End;

/《只是声明，不用实现，让包体实现，这是必须的流程》

1. 控制台调用：select 包.N(参数) from dual；
2. java调用：

sql=”select包. N(参数) from dual”;

ps=ct.prepareStatement(sql);

ps.setString(1,值);《循环给参数统一赋值》

i=execute();/rs=execute();

1. 《最后实现留给包体做》
2. 用存储过程来完成分页
3. 创建一个表的序列

--创建序列《从11开始，每次加1，最小值11，最大值不限，不循环，不缓存》

create sequence users\_s

start with 11

increment by 1

minvalue 11

nomaxvalue

nocycle

nocache

/

users\_s.nextval/ users\_s.currval《下一个序列值/当前的那个序列值》

1. 建一个包来放游标类型：

cearte package pack1 is

--声明游标类型

type cursor\_common is ref cursor;

end;

《只是声明，不用实现，让包体实现，这是必须的流程好像没有begin》

1. 编写分页的存储过程：《它自动会return区别于函数》

六个参数：

1、in\_table varchar2 in\_pageSize number 3、in\_pageNow numbr

4、out\_result pack1.cursor\_common 5、out\_rowCount number 6、out\_psgeCount number

V\_sql=’select t2.\* from (select t1.\*,rownum rn from (select\*from emp) t1 where rownum<=’||v\_end||’) t2 where rn>=’||v\_start||’ order by ‘||in\_empno;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create or replace procedure fenye(六个参数)

v\_sql varchar2(4000);

v\_startRow number:=in\_pageSize\*(in\_pageNow-1)+1;

v\_endRow number:=in\_pageSize\*pageNow;

begin

open out\_result for sql;

select count(\*) into out\_rowCount from emp;

out\_pageCount:=ceil(out\_rowCount/in\_pageSise);

end;

最后java调用：把游标转成rs对象，然后的流程和之前一样的开发你的java程序

注意《内层用\*的话如果两个表有重列的话就会报错，要把两个表的全部字段，一个个取出来，比如外键列和内键列重名就会报错》