Oracle 数据库

# 1 odbc 和jdbc

ODBC（Open Database Connectivity）是一组对数据库访问的标准API，这些API通过SQL来完成大部分任务，而且它本身也支持SQL语言，支持用户发来的SQL。ODBC定义了访问数据库API的一组规范，这些API独立于形色各异的DBMS和编程语言。

也就是说，一个基于ODBC的应用程序，对数据库的操作不依赖任何DBMS，不直接与DBMS打交道，所有的数据库操作由对应的DBMS的ODBC驱动程序完成。不论是SQL Server、Access还是Oracle数据库，均可用ODBC API进行访问。

由此可见，ODBC的最大优点是能以统一的方式处理所有的数据库。

JDBC（JavaDatabase Connectivity）是Java与数据库的接口规范，JDBC定义了一个支持标准SQL功能的通用低层API，它由Java 语言编写的类和接口组成，旨在让各数据库开发商为Java程序员提供标准的数据库API。

JDBC API定义了若干Java中的类，表示数据库连接、SQL指令、结果集、数据库元数据等。它允许Java程序员发送SQL指令并处理结果

**共同点** ：并不是直接来链接数据库 都是通过驱动进行来管理数据

一般都是四个组件 应用程序、驱动程序管理器、驱动程序和数据源,工作原 理亦大体相同；

**不同点**：Java 可以使用 ODBC，但最好是以JDBC-ODBC桥的形式使用（Java连接总体分为Java直连和JDBC-ODBC桥两种形式）。

那为什么还需要 JDBC？

因为ODBC 不适合直接在 Java 中使用，因为它使用 C 语言接口。从Java 调用本地 C代码在安全性、实现、坚固性和程序的自动移植性方面都有许多缺点。从 ODBC C API 到 Java API 的字面翻译是不可取的。例如，Java 没有指针，而 ODBC 却对指针用得很广泛（包括很容易出错的指针"void \*"）。

另外，ODBC 比较复杂，而JDBC 尽量保证简单功能的简便性，同时在必要时允许使用高级功能。如果使用ODBC，就必须手动地将 ODBC 驱动程序管理器和驱动程序安装在每台客户机上。如果完全用 Java 编写 JDBC 驱动程序则 JDBC代码在所有 Java 平台上（从网络计算机到大型机）都可以自 动安装、移植并保证安全性。

总之，JDBC 在很大程度上是借鉴了ODBC的，从他的基础上发展而来。JDBC 保留了 ODBC 的基本设计特征，因此，熟悉 ODBC 的程序员将发现 JDBC 很容易使用。它们之间最大的区别在于：JDBC 以 Java 风格与优点为基础并进行优化，因此更加易于使用。

## 1.1具体进行相应的连接操作

public class DbUtil {

public static Connection getConnection(){

Connection conn=null;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");//找到oracle驱动器所在的类

String url="jdbc:oracle:thin:@localhost:1521:bjpowernode"; //URL地址

String username="drp";

String password="drp";

conn=DriverManager.getConnection(url, username, password);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return conn;

}

<pre name="code" class="java">

public static void close(PreparedStatement pstmt){

if(pstmt !=null){

try {

pstmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void close(ResultSet rs){

if(rs !=null){

try {

rs.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

在运行Class.forName时，须要找到OracleDriver所在的地址，路径为：

1、首先找到

D:\oracle\product\10.2.0\db\_1\jdbc\lib 找到ojdbc14.jar

2、其次再找到 ojdbc14.jar\oracle\jdbc\driver 以下的oraceldriver这样就找到了要使用的驱动程序文件

2、操作数据库--加入

public void addUser(User user){

String sql="insert into t\_user(user\_id,user\_name,PASSWORD,CONTACT\_TEL,EMAIL,CREATE\_DATE)values(?

,?

,?,?,?

,?)"; //?为參数占位符

Connection conn=null;

PreparedStatement pstmt=null; //通常利用PreparedStatement进行操作，性能得到优化

try{

conn=DbUtil.getConnection();

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, user.getUserId());

pstmt.setString(2,user.getUserName());

pstmt.setString(3, user.getPassword());

pstmt.setString(4, user.getContactTel());

pstmt.setString(5,user.getEmail());

//pstmt.setTimestamp(6,new Timestamp(System.currentTimeMillis()));

pstmt.setTimestamp(6, new Timestamp(new Date().getTime()));//获取当前系统时间

pstmt.executeUpdate();//运行增删改操作

}catch(SQLException e){

e.printStackTrace();

}finally{

DbUtil.close(conn);

DbUtil.close(pstmt);

}

}

## 1.2操作数据库--查询

public User findUserById(String userId){

String sql = "select user\_id, user\_name, password, contact\_tel, email, create\_date from t\_user where user\_id=?";

Connection conn=null;

PreparedStatement pstmt=null;

ResultSet rs=null;//定义存放查询结果的结果集

User user=null;

try{

conn=DbUtil.getConnection();

pstmt=conn.prepareStatement(sql);

pstmt.setString(1,userId);

rs=pstmt.executeQuery();//运行查询操作

if(rs.next()){

user=new User();

user.setUserId(rs.getString("user\_Id"));

user.setUserName(rs.getString("user\_name"));

user.setPassword(rs.getString("password"));

user.setContactTel(rs.getString("contact\_Tel"));

user.setEmail(rs.getString("email"));

user.setCreateDate(rs.getTimestamp("create\_date"));

}

}catch(SQLException e){

e.printStackTrace();

}finally{

//按顺序进行关闭

DbUtil.close(rs);

DbUtil.close(pstmt);

DbUtil.close(conn);

}

return user;

}

1、PreparedStatement与Statement的简单差别

Statement为一条SQL语句生成运行计划，假设參数值不同。会生成不同的sql语句，运行相应參数值个数的次数。假设仅仅有一条SQL语句运行时，最好採用Statement进行。

PreparedStatement使用绑定变量重用运行计划。不同的參数值相应的查询语句。仅仅会生成一个sql语句。大大提高了运行的效率。是预编译的。在大批量语句操作时，提高了效率，同一时候可採用'?'来代表參数，能够防止SQL注入。安全性更高。

2、与之前的知识进行联系

眼下接触到的额JDBC中的连接对象有Connection这个与之前ODBC的Connection作用同样，为数据库连接对象。而PraparedStatement与Statement与ODBC中的command对象类似。都是用来运行SQL语句的。

查询方法中用到的ResultSet则与之前用到的DataSet或者DataTable功能类似，

# 2 Oracle 的基本配置

去官网下载oracle11

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>

2可视化界面

PLsql developer 下载链接<https://www.allroundautomations.com/plsqldev.html>

# 3 表空间

扩展表空间的方法举例：

1.创建一个表空间ts01

create tablespace ts01 datafile 'D:\test\ts01.dbf' size 1m;

create tablespace ts01 datafile 'D:\test\ts01.dbf' size 1m reuse;

2.在该表空间上创建一个t表，其结构和dept表一样

create table t(no number(2),name varchar2(14),loc varchar2(13)) tablespace ts01;

3.向t表中插入数据

insert into t select \* from scott.dept;

insert into t select \* from t;

4.扩展表空间

（1）增加数据文件

alter tablespace ts01 add datafile 'D:\test\ts02.dbf' size 1m;

（2）增加数据文件大小

alter database datafile 'D:\test\ts01.dbf' resize 6m;

（3）设置数据文件的大小为自动增长

alter database datafile 'D:\test\ts02.dbf' autoextend on next 5m maxsize 50m;

（4）增加数据文件并设置自动增长

alter tablespace ts01 add datafile 'D:\test\ts04.dbf' size 2m autoextend on next 5m maxsize 50m;

删除表空间ts01

drop tablespace ts01 including contents and datafiles;或者:

drop tablespace ts01 including contents and datafiles cascade constraints;

删除表空间的数据文件ts02.dbf

alter tablespace ts01 drop datafile 'D:\test\ts02.dbf';

# 4 权限回收 验证

4.1系统权限回收：

如果数据库管理员用GRANT 命令给用户A授予系统权限时带有WITH ADMIN OPTION 选项，则用户A有权将系统权限再次授予另外的用户B。在这种情况下，如果数据库管理员使用REVOKE命令撤销A用户的系统权限，用户B的系统权限是否存在？(存在)

举例：system(create session)->a(create session)->b

如果a的系统权限被回收，那么b的系统权限是否被级联回收？

验证过程：

conn system/orcl;

create user a identified by a;

create user b identified by b;

grant create session to a with admin option;

conn a/a;

grant create session to b;

conn b/b;

conn system/orcl;

revoke create session from a;

conn b/b;

4.2对象权限回收：

如果数据库管理员用GRANT 命令给用户A授予对象权限时带有WITH GRANT OPTION 选项，则用户A有权将对象权限再次授予另外的用户B。在这种情况下，如果数据库管理员使用REVOKE命令撤销A用户的对象权限，用户B的对象权限是否存在？()

举例：system(查询emp)->a(查询emp)->b

如果a的对象权限被回收，那么b的对象权限是否被级联回收？

验证过程：

conn system/orcl;

grant create session to a;

grant select on scott.emp to a with grant option;

conn a/a;

select \* from scott.emp;

grant select on scott.emp to b;

conn b/b;

select \* from scott.emp;

conn system/orcl;

revoke select on scott.emp from a;

conn a/a;

select \* from scott.emp;

conn b/b;

select \* from scott.emp;

# 5表结构和表数据

1.通过本地化管理方式创建一个表空间ts01，其扩展大小为自动管理，其段空间管理方式为自动。

create tablespace ts01

datafile 'D:\test\ts01.dbf'

size 1m reuse

extent management local autoallocate

segment space management auto;

2.创建一个用户a,密码也是a,其默认的表空间为ts01。限制a用户可使用ts01上20m的限额。

create user a identified by a

default tablespace ts01

quota 10m on ts01;

3.给用户a授予connect和resource权限。使用用户a登录，创建一个表table1 结构如下：

grant connect,resource to a;

conn a/a;

create table table1(no number(2),name varchar2(14),loc varchar2(13));

4.将表table1的表名称修改为department

alter table table1 rename to department;

5.给表增加一个字段 name2 varchar2(20)

alter table department add(name2 varchar2(20));

6.删除表中的name2字段

alter table department drop column name2;

7.修改表中loc字段名为地址address

alter table department rename column loc to address;

8.可否将字段名address改为 add ?

SQL> alter table department rename column address to add;

alter table department rename column address to add

ORA-00904: : 标识符无效

9.修改表中的字段no 的数据类型为char(20)不为空。

alter table department modify no char(20) not null;

10.给表增加一个主键约束（以no 字段为主键）

alter table department

add constraint pk\_department primary key(no);

11.向表中插入部门编号为1的一条数据

insert into department (no) values(1);

12.向表中插入一条数据:部门编号为2，名称为SALES，地址为 BOSTON

insert into department values(2,'sales','boston');

13.将部门编号为1的记录，的部门名称更新为 new york

update department set name='new york' where no=1;

14.将部门编号为1的记录删除

delete from department where no=1;

# 6 表的查询和高级查询

（1）查询选修课程‘3-105’且成绩在60到80之间的所有记录。

select \* from XS\_KC where grade between 60 and 80 and cno ='3-105';

NO CNO GRADE

---- ---------- -----

109 3-105 76

101 3-105 64

108 3-105 78

（2）查询成绩为85、86或88的记录。

select \* from XS\_KC where grade='85' or grade='86' or grade='88';

NO CNO GRADE

---- ---------- -----

103 3-245 86

105 3-105 88

101 6-166 85

（3）查询95031班的学生人数。

select count(no) from XS where class = '95031';

COUNT(NO)

----------

3

（4）查询至少有5名学生选修并以3开头的课程的平均成绩。

select avg(grade) from XS\_KC where group by cno having count(\*)>5 and e cno like '3-%'

（5）查询最低分大于70，最高分小于90的NO列。

select no from XS\_KC where grade between 70 and 90;

NO

----

103

105

105

109

108

101

107

108

8 rows selected

（6）查询95033班学生所选课程的平均分。

select avg(grade) from XS\_KC,XS where XS.no = XS\_KC.no and class='95033';

AVG(GRADE)

----------

79.6666666

（7）查询选修‘3-105’课程的且成绩高于‘109’号同学成绩的所有同学的记录。

select xs.no,xs.name,xs\_kc.grade

from xs st,xs\_kc sc

where

sc.cno='3-105' and sc.grade>(select grade

from xs\_kc

where cno='3-105'and no='109')

and st.no=sc.no;

（8）查询与学号为108的同学同岁的所有学生的NO、NAME和AGE。

select no,name, age from XS where age = (select age from XS where no = '108');

NO NAME AGE

---- ---------- ---

108 曾华 19

101 李军 19

（9）查询“张旭”教师任课的课程号，选修其课程学生的学号和成绩。

SQL> select tno from TC where name='张旭';

TNO

----

856

SQL> select no ,grade from XS\_KC ,TC,KC where TC.name = '张旭' and TC.tno = KC.tno and KC.cno = XS\_KC.cno;

NO GRADE

---- -----

108 81

107 79

101 85

（10）查询选修其课程的学生人数多于5人的教师姓名。

SQL> select TC.name,XS\_KC.cno from XS\_KC ,TC,KC where TC.tno = KC.tno and KC.cno = XS\_KC.cno group by XS\_KC.cno,TC.name having count(\*)>5;

NAME CNO

---------- ----------

王萍 3-105

（11）查询“计算机系”与“电子工程系”不同职称的教师的姓名和职称。

SQL> select name,prof

2 from tc

3 where depart='计算机系' or depart='电子工程'

4 group by prof,name;

NAME PROF

---------- ----------

刘冰 助教

张旭 讲师

李城 副教授

王萍 助教

（12）查询选修编号为“3-105”课程且成绩至少高于选修编号为“3-245”课程的同学的CNO、NO 、GRADE并按GRADE从高到低次序排列。

SQL> select \*

2 from XS\_KC

3 where cno='3-105' and grade > all (select grade

4 from XS\_KC

5 where cno='3-245') order by grade desc;

NO CNO GRADE

---- ---------- -----

103 3-105 92

107 3-105 91

105 3-105 88

（13）查询选修编号为“3-105”课程且成绩高于选修编号为“3-245”课程的同学的CNO、NO 、GRADE。

SQL> select \*

2 from xs\_kc

3 where cno='3-105' and grade > all (select grade

4 from xs\_kc

5 where cno='3-245');

NO CNO GRADE

---- ---------- -----

105 3-105 88

107 3-105 91

103 3-105 92

（14）列出所有教师和同学的NAME 、SEX 、AGE。

SQL> select xs.name,xs.sex,xs.age,

2 tc.name,tc.sex,tc.age

3 from

4 xs inner join (xs\_kc inner join (kc inner join tc on kc.tno=tc.tno)

5 on xs\_kc.cno=kc.cno) on xs.no=xs\_kc.no;

NAME SEX AGE NAME SEX AGE

---------- --- --- ---------- --- ---

曾华 男 19 张旭 男 35

曾华 男 19 王萍 女 28

匡明 男 20 李城 男 42

匡明 男 20 王萍 女 28

王丽 女 20 张旭 男 35

王丽 女 20 王萍 女 28

李军 男 19 张旭 男 35

李军 男 19 王萍 女 28

王芳 女 22 李城 男 42

王芳 女 22 王萍 女 28

陆君 男 20 李城 男 42

陆君 男 20 王萍 女 28

12 rows selected

（15）查询成绩比该课程平均成绩低的学生的成绩表。

SQL> select \*

2 from xs\_kc

3 where grade < any (select avg(grade) from xs\_kc group by cno) order by cno asc;

NO CNO GRADE

---- ---------- -----

101 3-105 64

108 3-105 78

109 3-105 76

109 3-245 68

105 3-245 75

108 6-166 81

107 6-166 79

7 rows selected

（16）列出所有任课教师的NAME和DEPART。

SQL> select name,depart

2 from tc;

NAME DEPART

---------- --------------------

李城 计算机系

张旭 电子工程

王萍 计算机系

刘冰 电子工程

（17）列出所有未讲课教师的NAME和DEPART。

SQL> select tc.name as 教师姓名,depart as 所属院系

2 from tc

3 where tno not in

4 (select tno

5 from kc

6 where cno in (

7 select cno

8 from xs\_kc group by cno));

教师姓名 所属院系

---------- --------------------

刘冰 电子工程

（18）列出至少有2名男生的班号。

SQL> select class as 班号

2 from xs

3 where sex='男' group by class having count(class) >= 2;

班号

-----

95033

95031

（19）查询不姓“王”的学生记录。

SQL> select \*

2 from xs

3 where name not in (select name from xs where name like '王%');

NO NAME SEX AGE CLASS

---- ---------- --- --- -----

105 匡明 男 20 95031

103 陆君 男 20 95031

108 曾华 男 19 95033

101 李军 男 19 95033

（20）查询每门课最高分的学生的NO、CNO、GRADE。

SQL> select \*

2 from xs\_kc

3 where grade in (select max(grade) from xs\_kc group by cno);

NO CNO GRADE

---- ---------- -----

103 3-245 86

103 3-105 92

101 6-166 85

（21）查询与“李军”同性别并同班的同学NAME。

SQL> select name

2 from xs

3 where sex in (select sex

4 from xs

5 where name='李军') and class in (select class

6 from xs

7 where name='李军');

NAME

----------

曾华

李军

（22）查询“男”教师及其所上的课程。

SQL> select name,cname

2 from tc,kc

3 where sex='男' and tc.tno=kc.tno;

NAME CNAME

---------- --------------------

李城 操作系统

张旭 数字电路

（23）查询选修“计算机导论”课程的“男”同学的成绩表。

SQL> select xs.no,xs.name,grade

2 from xs,kc,xs\_kc

3 where xs.sex='男' and kc.cname='计算机导论' and kc.cno=xs\_kc.cno

4 and xs.no=xs\_kc.no;

NO NAME GRADE

---- ---------- -----

108 曾华 78

105 匡明 88

101 李军 64

103 陆君 92

（24）往基本表Student中插入一个学生元组（'张红'，'女'）。

SQL> insert into xs

2 values('110','张红','女',18,'95031');

1 row inserted

（25）在基本表xs\_kc中删除尚无成绩的选课元组。

SQL> delete from XS\_KC where grade =null;

0 rows deleted

（26）把选修数学课不及格的成绩全改为空值。

SQL> update xs\_kc set grade =null where grade<60 and cno in (select cno from kc where cname='高等数学');

0 rows updated

（27）查询分数低于总平均成绩的女同学的姓名，并把其成绩提高5%。

SQL> select name,grade\*1.05

2 from xs\_kc,xs

3 where xs\_kc.no=xs.no and sex='女' and grade<(select avg(grade)

4 from xs\_kc);

NAME GRADE\*1.05

---------- ----------

王芳 71.4

王芳 79.8

王丽 82.95

（28）查询“张旭”教师任课的课程号，选修其课程学生的学号和成绩。

SQL> select kc.cno,xs\_kc.no,xs\_kc.grade

2 from kc,xs\_kc,tc

3 where tc.name='张旭' and tc.tno=kc.tno and xs\_kc.cno=kc.cno;

CNO NO GRADE

---------- ---- -----

6-166 108 81

6-166 107 79

6-166 101 85

（29）检索学号比李同学大，而年龄比他小的学生姓名。

SQL> select name,no,age

2 from xs

3 where no>(select no from xs where name like '李%')and

4 age<(select age from xs where name like '李%');

NAME NO AGE

---------- ---- ---

张红 110 18

# 7 数据库的连接和分页查询

## 7.1数据库的连接和存储过程的调用

package com.hcm;

import java.sql.\*;

public class Test {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection ct=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:orcl","scott","orcl");

CallableStatement cs=ct.prepareCall("{call fenye(?,?,?,?,?,?)}");

//给?赋值

cs.setString(1, "emp");

cs.setInt(2, 5);

cs.setInt(3, 1);

//注册总记录数

cs.registerOutParameter(4, oracle.jdbc.OracleTypes.INTEGER);

//注册总页数

cs.registerOutParameter(5,oracle.jdbc.OracleTypes.INTEGER);

//注册返回的结果集

cs.registerOutParameter(6, oracle.jdbc.OracleTypes.CURSOR);

cs.execute();

//取出总记录数/这里要注意，getInt(4)中的4是由该参数的位置决定的

int rowNum=cs.getInt(4);

int pageCount=cs.getInt(5);

ResultSet rs=(ResultSet)cs.getObject(6);

//显示一下，看对不对

System.out.println("rowNum="+rowNum);

System.out.println("总页数："+pageCount);

while(rs.next()){

System.out.println("编号："+rs.getInt(1)+"名字："+rs.getString(2)+"薪水："+rs.getDouble(6));

}

} catch (Exception e) {

e.printStackTrace();

// TODO: handle exception

}

}

}

## 7.2具体的分页存储过程

1、先开发一个包，在该包中，定义类型test\_cursor,是个游标。

create or replace package testpackage as

type test\_cursor is ref cursor;

end testpackage;

2、编写分页的过程,要求可以输入表名、每页显示记录数、当前页。返回总记录数，总页数，和返回的结果集。

create or replace procedure fenye

(tableName in varchar2,

pageSize in number,--每页显示记录数

pageNow in number,

myrows out number,--总记录数

pageCount out number,--总页数

p\_cursor out testpackage.test\_cursor --返回的记录集

) is

--定义部分

--定义sql语句 字符串

var\_sql varchar2(1000);

--定义两个整数

var\_begin number:=(pageNow-1)\*pageSize+1;

var\_end number:=pageNow\*pageSize;

begin

--执行部分

var\_sql:='select \* from (select a1.\*,rownum rn from (select \* from '|| tableName ||' order by sal) a1 where rownum<='|| var\_end ||') where rn>='|| var\_begin;

--把游标和sql关联

open p\_cursor for var\_sql;

--计算myrows和pageCount

--组织一个sql

var\_sql:='select count(\*) from '||tableName;

--执行sql,并把返回的值，赋给myrows;

execute immediate var\_sql into myrows;

--计算pageCount

if mod(myrows,pageSize)=0 then

pageCount:=myrows/pageSize;

else

pageCount:=myrows/pageSize+1;

end if;

--关闭游标??

--close p\_cursor;

end;

/

3、使用java测试。要求按照工资从低到高排序，然后取出第6-10个记录的编号、姓名、工资。