oracleOCCI的一个简单的包装类的实现 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/214/2021_2022_oracleOCCI_c102_214464.htm 最近在学习oracle 的c 的编程接口OCCI，自己做了一个简单此程序并没有经过严格的测试，代码在vs2005和AIX的xlC中测试通过。 注意：如果需要在vs2005中链接，需要到oracle网站上下载最新的vs2005的occi库文件。以下是引用片段： TOcci.h #ifndef _OCCIDATABASE_H_ #define _OCCIDATABASE_H_ #include #include #include using namespace oracle::occi. using namespace std. namespace happyever { class TOcciDatabase { public: static TOcciDatabase* getInstance(string usr, string passwd, string db). int getConnectCount(){ return _Instance->count. }. Connection* getConnect(){ count .return _Instance->conn. }. ~TOcciDatabase(). protected: TOcciDatabase(){}. TOcciDatabase(string usr, string passwd, string db). private: static TOcciDatabase* _Instance. static int count. Environment *env. Connection *conn. }. int TOcciDatabase::count = 0. TOcciDatabase* TOcciDatabase::_Instance = 0. TOcciDatabase::TOcciDatabase(string usr, string passwd, string db) { try { env = Environment::createEnvironment (Environment::DEFAULT). conn = env->createConnection (usr, passwd, db). } catch(SQLException ex) { cout cout cout throw ex. } }. TOcciDatabase::~TOcciDatabase() { try { env->terminateConnection (conn). Environment::terminateEnvironment (env). } catch(SQLException ex) { cout cout cout throw ex. } }. TOcciDatabase* TOcciDatabase::getInstance(string usr, string passwd, string db) { if(_Instance == 0) { _Instance = new TOcciDatabase(usr,passwd,db). } return _Instance. }. class TOcciQuery { private: Connection *conn. Statement *stmt. bool isAutoCommit. TOcciQuery(){}. public : TOcciQuery(Connection *connect){ conn = connect. }. void beginTrans(). void commit(). void roolback(). boolean getAutoCommit(). ResultSet* executeQuery(string sql) . void executeUpdate(string sql) . void close() { if(stmt != NULL) conn->terminateStatement (stmt). }. void close(ResultSet* rs). }. void TOcciQuery::close(ResultSet* rs) { if(rs != NULL) stmt->closeResultSet (rs). if(stmt != NULL) conn->terminateStatement (stmt). }. void TOcciQuery::beginTrans() { try { isAutoCommit = stmt->getAutoCommit(). stmt->setAutoCommit(false). } catch(SQLException ex) { cout cout cout throw ex. } }. void TOcciQuery::commit() { try { conn->commit(). stmt->setAutoCommit(isAutoCommit). } catch(SQLException ex) { cout cout cout throw ex. } }. void TOcciQuery::roolback() { try { conn->rollback(). stmt->setAutoCommit(isAutoCommit). } catch(SQLException ex) { cout cout cout throw ex. } }. boolean TOcciQuery::getAutoCommit() { boolean result = false. try { result = stmt->getAutoCommit(). } catch(SQLException ex) { cout cout cout throw ex. } return result. }. ResultSet* TOcciQuery::executeQuery(string sql) { ResultSet*rs = NULL. try { stmt = conn->createStatement(). rs = stmt->executeQuery(sql). }

catch (SQLException ex) { cout cout cout throw ex. } return rs. }. void TOcciQuery::executeUpdate(string sql) { try { stmt = conn->createStatement(). stmt->executeUpdate(sql). } catch (SQLException ex) { cout cout cout throw ex. } }. } #endif /*_OCCIDATABASE_H_*/ 测试程序main.cpp源码如下： // occi.cpp : 定义控制台应用程序的入口点。 // #include "stdafx.h" #include "TOcci.h" int _tmain(int argc, _TCHAR* argv[]) { using namespace happyever. TOcciQuery *query = new TOcciQuery(TOcciDatabase::getInstance("cal","cal","v2b76")->getConnect()). string strSQL = "0select count(*) from serv_value_total". ResultSet* rs = query->executeQuery(strSQL). while(rs->next()) { std::cout } query->close(rs). 0delete(query). return 1. }