

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的个人博客项目的数据库设计 | | |
| 课程 | Oracle数据库应用 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 赵张阳 | 学号 | 201810414329 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 10 |  |
| 表设计 | 表，表空间设计合理，数据合理 | 20 |  |
| 用户管理 | 权限及用户分配方案设计正确 | 20 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 30 |  |
| 备份方案 | 备份方案设计正确 | 20 |  |
| **得分合计** | | |  |

2021 年 6 月 1 日

目录

[**期末考核要求** 2](#_Toc74569837)

[1.项目简介 3](#_Toc74569838)

[2.功能分析 3](#_Toc74569839)

[3.E-R图设计 4](#_Toc74569840)

[4.类图设计 5](#_Toc74569841)

[5.数据库设计 5](#_Toc74569842)

[5.1 User表 5](#_Toc74569843)

[5.2 Friends表 6](#_Toc74569844)

[5.3 分类表 6](#_Toc74569845)

[5.4 博文表 7](#_Toc74569846)

[5.5 评论表 7](#_Toc74569847)

[5.6 标签表 8](#_Toc74569848)

[6.项目流程 8](#_Toc74569849)

[6.1创建表空间 8](#_Toc74569850)

[6.2创建表 8](#_Toc74569851)

[6.3添加数据 11](#_Toc74569852)

[6.4设计权限及用户分配方案,两类角色，两个用户 15](#_Toc74569853)

[6.5存储过程和函数 16](#_Toc74569854)

[6.6自动备份方案 16](#_Toc74569855)

[6.6.1新建 backup.bat脚本 17](#_Toc74569856)

[6.6.2创建 windows任务计划 18](#_Toc74569857)

[6.6.3编写拷贝程序 21](#_Toc74569858)

[6.6.4 编写拷贝脚本 27](#_Toc74569859)

[6.6.5设定windows任务计划 27](#_Toc74569860)

[7.项目总结 30](#_Toc74569861)

**期末考核要求**

* 自行设计一个信息系统的数据库项目，自拟某项目名称。
* 设计项目涉及的表及表空间使用方案。至少5张表和5万条数据，两个表空间。
* 设计权限及用户分配方案。至少两类角色，两个用户。
* 在数据库中建立一个程序包，在包中用PL/SQL语言设计一些存储过程和函数，实现比较复杂的业务逻辑，用模拟数据进行执行计划分析。
* 设计自动备份方案或则手工备份方案。

# 1.项目简介

项目名称：个人博客系统

本项目是基于Oracle的个人博客系统数据库设计。

涉及角色/用户：普通用户、游客

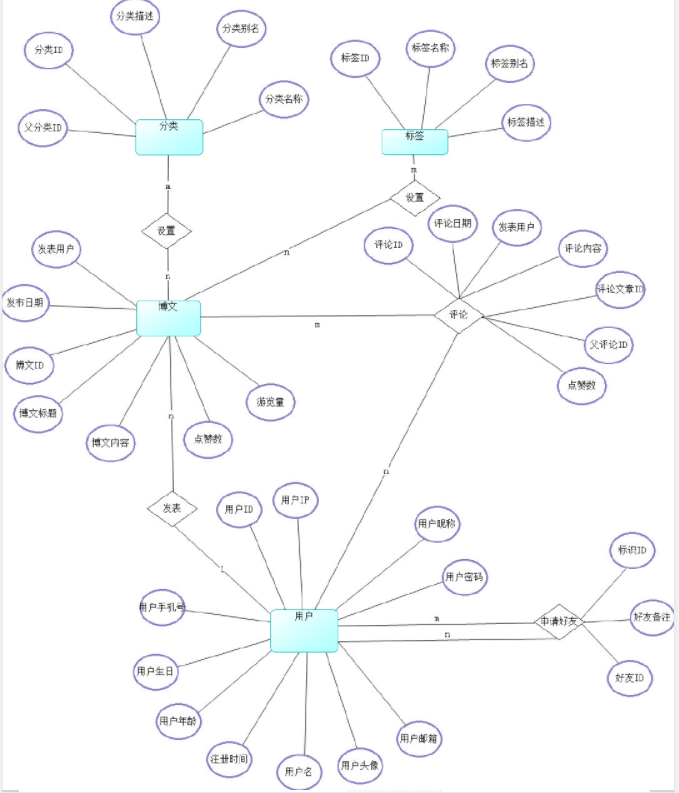
涉及表：用户表、游客表、朋友表、文章标签表、文章分类表、内容表、评论表

# 2.功能分析

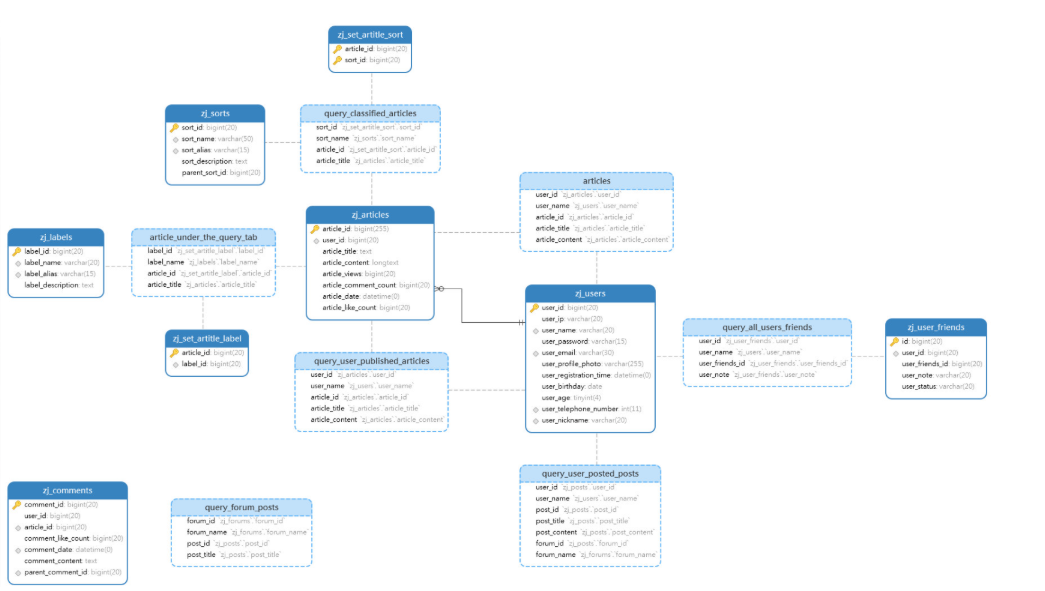
系统主要完成以下几方面的功能：

* 用户管理：用户的注册和登录，发表博文和评论。
* 博文管理：用户可以在网站中发表和设置博文。
* 评论管理：用户可以评论博文和回复其他用户的评论。
* 分类管理：添加和删除分类，给文章设置分类。
* 标签管理：添加和删除标签，给文章设置标签。

# 3.E-R图设计



# 4.类图设计



# 5.数据库设计

## 5.1 User表



## 5.2 Friends表



## 5.3 分类表



## 5.4 博文表



## 5.5 评论表



## 5.6 标签表



# 6.项目流程

## 6.1创建表空间

CREATE TABLESPACE Users02

DATAFILE

'/home/oracle/app/oracle/oradata/orcl/pdborcl/date01.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/date02.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

## 6.2创建表

-- user表的创建

create table manage

(

user\_id number(20) primary key,

user\_ip VARCHAR2(20 BYTE） not null,

user\_name VARCHAR2(100 BYTE） not null,

user\_password VARCHAR2(20 BYTE）not null,

user\_email VARCHAR2(100 BYTE） not null,

user\_profile\_photo VARCHAR2(255 BYTE） not null,

user\_registration\_time datetime not null,

user\_birthday date not null,

user\_age number(20) not null,

user\_telephone\_number number(20) not null,

user\_nickname VARCHAR2(100 BYTE）

)

tablespace date01

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

-- Friends表的创建

create table Friends

(

id number(20) primary key,

user\_id number(20) not null,

user\_friends\_id number(20) not null,

user\_note VARCHAR2(100 BYTE） not null,

user\_status VARCHAR2(100 BYTE）

)

tablespace date01

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

-- 分类表的创建

create table sort

(

sort\_id number(20) primary key,

sort\_name VARCHAR2(100 BYTE） not null,

sort\_alias VARCHAR2(100 BYTE） not null,

sort\_description VARCHAR2(255 BYTE） not null,

parrent\_sort\_id number(20)

)

tablespace date02

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

-- 博文表的创建

create table article

(

article\_id number(20) primary key,

user\_id number(20) not null,

article\_title VARCHAR2(255 BYTE） not null,

article\_content VARCHAR2(100 BYTE） not null,

article\_views number(20) not null,

article\_comment\_count number(20) not null,

artcle\_date DATETIME not null,

article\_like\_count number(20)

)

tablespace date02

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

-- 评论表的创建

create table comment

(

comment\_id number(20) primary key,

user\_id number(20) not null,

comment\_id number(20) not null,

comment\_like\_count number(20) not null,

comment\_date DATETIME not null,

comment\_content VARCHAR2(255 BYTE） not null,

parent\_comment\_id number(20)

)

tablespace date02

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

-- 标签表的创建

create table label

(

label\_id number(20) primary key,

label\_name VARCHAR2(255 BYTE） not null,

label\_alias VARCHAR2(100 BYTE） not null,

label\_description VARCHAR2(255 BYTE） not null

)

tablespace date02

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

## 6.3添加数据

-- user表的数据插入20000条

declare

i int;

user\_id number(20) primary key;

user\_ip VARCHAR2(20 BYTE） not null;

user\_name VARCHAR2(100 BYTE） not null;

user\_password VARCHAR2(20 BYTE）not null;

user\_email VARCHAR2(100 BYTE） not null;

user\_profile\_photo VARCHAR2(255 BYTE） not null;

user\_registration\_time datetime not null;

user\_birthday date not null;

user\_age number(20) not null;

user\_telephone\_number number(20) not null;

user\_nickname VARCHAR2(100 BYTE）;

begin

i:=1;

while i<=20000

loop

user\_id:=i;

user\_ip:=i;

user\_name:= 'user'|| i;

user\_password := '123'|| i;

user\_email := '111222333'|| i;

user\_profile\_photo :='photo'||i;

user\_registration\_time :='time'||i;

user\_birthday :='birthday'||i;

user\_age := '18'|| i;

user\_telephone\_number := '1808494'|| i;

user\_nickname := 'fengsheng'||i

insert into user(USER\_ID,USER\_ID,USER\_NAME,USER\_PASSWORD,USER\_EMAIL,USER\_PROFILE\_PHOTO,USER\_REGISTRATION\_TIME,USER\_BIRTHDAY,USER\_AGE,USER\_TELEPHONT\_NUMBER,USER\_NICKNAME) values (user\_id,user\_ip,user\_name,user\_password,user\_email,user\_profile\_photo,user\_registration\_time,user\_birthday,user\_age,user\_telephone\_number,user\_nickname);

i:=i+1;

end loop;

commit;

end;

/

-- friends表的数据插入20000条

declare

i int;

id number(20) primary key;

user\_id number(20) not null;

user\_friends\_id number(20) not null;

user\_note VARCHAR2(100 BYTE） not null;

user\_status VARCHAR2(100 BYTE);

begin

i:=1;

while i<=20000

loop

id:=i;

user\_id:='用户id'||i;

user\_friends\_id:= '好友id'|| i;

user\_note := '备注'|| i;

user\_status := '状态'|| i;

insert into user(ID,USER\_ID,USER\_FRIENDS\_ID,USER\_NOTE,USER\_STATUS) values (id,user\_id,user\_friends\_id,user\_note,user\_status);

i:=i+1;

end loop;

commit;

end;

/

-- 分类表的数据插入20000条

declare

i int;

sort\_id number(20) primary key6;

sort\_name VARCHAR2(100 BYTE） not null;

sort\_alias VARCHAR2(100 BYTE） not null;

sort\_description VARCHAR2(255 BYTE） not null;

parrent\_sort\_id number(20);

begin

i:=1;

while i<=20000

loop

sort\_id:=i;

sort\_name:='分类名称'||i;

sort\_alias:= '分类别名'|| i;

sort\_description := '分类描述'|| i;

parrent\_sort\_id := '父分类id'|| i;

insert into user(ID,USER\_ID,USER\_FRIENDS\_ID,USER\_NOTE,USER\_STATUS) values (id,user\_id,user\_friends\_id,user\_note,user\_status);

i:=i+1;

end loop;

commit;

end;

/

-- 博文表的数据插入10000条

declare

i int;

article\_id number(20) primary key;

user\_id number(20) not null;

article\_title VARCHAR2(255 BYTE） not null;

article\_content VARCHAR2(100 BYTE） not null;

article\_views number(20) not null;

article\_comment\_count number(20) not null;

artcle\_date DATETIME not null;

article\_like\_count number(20);

begin

i:=1;

while i<=10000

loop

article\_id:=i;

user\_id:=i;

article\_title:='博文标题'||i;

article\_content:='博文内容'||i;

article\_views:= i;

article\_comment\_count := i;

artcle\_date:=to\_date('2019-7-2','yyyy-mm-dd')+(i mod 60);

article\_like\_count:= i;

insert into user(ARTICLE\_ID,USER\_ID,ARTICLE\_TITLE,ARTICLE\_CONTENT,ARTICLE\_VIEWS,ARTICLE\_COMMENT\_COUNTARTICLE\_DATE,ARTICLE\_LIKE\_COUNT) values (article\_id,user\_id,article\_title,article\_content,article\_views,article\_comment\_count,artcle\_date,article\_like\_count);

i:=i+1;

end loop;

commit;

end;

/

-- 评论表的数据插入10000条

declare

i int;

comment\_id number(20) primary key;

user\_id number(20) not null;

comment\_id number(20) not null;

comment\_like\_count number(20) not null;

comment\_date DATETIME not null;

comment\_content VARCHAR2(255 BYTE） not null;

parent\_comment\_id number(20);

begin

i:=1;

while i<=10000

loop

comment\_id:=i;

user\_id:=i;

comment\_id:=i;

comment\_like\_count:=i;

comment\_date:=to\_date('2020-7-12','yyyy-mm-dd')+(i mod 60);

comment\_content:='评论内容'||i;

parent\_comment\_id:=i;

insert into user(COMMENT\_ID,USER\_ID,COMMENT\_ID,COMMENT\_LIKE\_COUNT,COMMENT\_DATE,COMMENT\_CONTENT,PARENT\_COMMENT\_ID) values (comment\_id,user\_id,comment\_id,comment\_like\_count,comment\_date,comment\_content,parent\_comment\_id);

i:=i+1;

end loop;

commit;

end;

/

-- 标签表的数据插入10000条

declare

i int;

label\_id number(20) primary key;

label\_name VARCHAR2(255 BYTE） not null;

label\_alias VARCHAR2(100 BYTE） not null;

label\_description VARCHAR2(255 BYTE） not null;

begin

i:=1;

while i<=10000

loop

label\_id:=i;

label\_name:='标签名'||i;

label\_alias:='标签别名'||i;

label\_description:='标签描述'i;

insert into user(LABEL\_ID,LABEL\_NAME,LABEL\_ALIAS,LABEL\_DESCRIPTION) values (label\_id,label\_name,label\_alias,label\_description);

i:=i+1;

end loop;

commit;

end;

/

## 6.4设计权限及用户分配方案,两类角色，两个用户

-- 创建角色

CREATE ROLE Visitor;

CREATE ROLE User;

-- 授权角色

GRANT connect,resource,dba, create table,create view,create trigger, create procedure,create sequence TO user

GRANT connect,resource, create table,create view,create trigger, create procedure,create sequence TO visitor;

-- 创建用户

CREATE Visitor visitor IDENTIFIED BY 123 DEFAULT TABLESPACE date01;

-- 创建用户

CREATE USER user IDENTIFIED BY 123 DEFAULT TABLESPACE date01,date02;

-- 分配角色给用户

GRANT Visitor TO visitor;

GRANT User TO user;

## 6.5存储过程和函数

create or replace PACKAGE BODY MyPack IS

FUNCTION Get\_SaleAmount(V\_DEPARTMENT\_ID NUMBER) RETURN NUMBER

AS

N NUMBER(20,2);

BEGIN

SELECT SUM(O.TRADE\_RECEIVABLE) into N FROM ORDERS O,EMPLOYEES E

WHERE O.EMPLOYEE\_ID=E.EMPLOYEE\_ID AND E.DEPARTMENT\_ID =V\_DEPARTMENT\_ID;

RETURN N;

END;

PROCEDURE GET\_EMPLOYEES(V\_EMPLOYEE\_ID NUMBER)

AS

LEFTSPACE VARCHAR(2000);

begin

--通过LEVEL判断递归的级别

LEFTSPACE:=' ';

--使用游标

for v in

(SELECT LEVEL,EMPLOYEE\_ID,NAME,MANAGER\_ID FROM employees

START WITH EMPLOYEE\_ID = V\_EMPLOYEE\_ID

CONNECT BY PRIOR EMPLOYEE\_ID = MANAGER\_ID)

LOOP

DBMS\_OUTPUT.PUT\_LINE(LPAD(LEFTSPACE,(V.LEVEL-1)\*4,' ')||

V.EMPLOYEE\_ID||' '||v.NAME);

END LOOP;

END;

END MyPack;

## 6.6自动备份方案

6.6.1新建 backup.bat脚本

@echo off

echo ================================================

echo Windows环境下Oracle数据库的自动备份脚本

echo 1. 使用当前日期命名备份文件。

echo ================================================

::以“YYYYMMDD”格式取出当前时间。

set BACKUPDATE=%date:~0,4%%date:~5,2%%date:~8,2%

::设置用户名、密码和要备份的数据库。

set USER=用户名

set PASSWORD=密码

set DATABASE=数据库实例名

::创建备份目录。

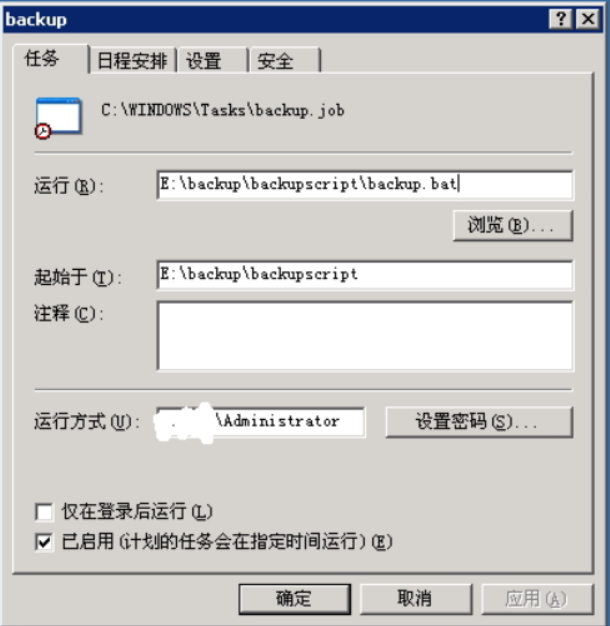
if not exist "E:\backup\backuptempdir" mkdir E:\backup\backuptempdir

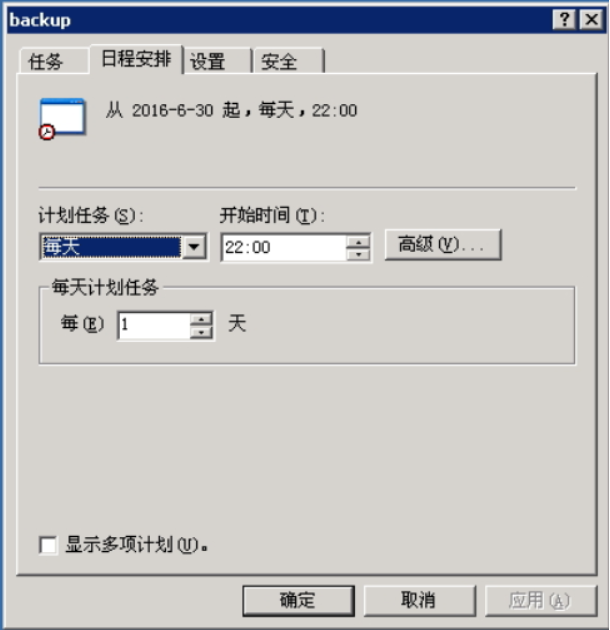
set DATADIR=E:\backup\backuptempdir

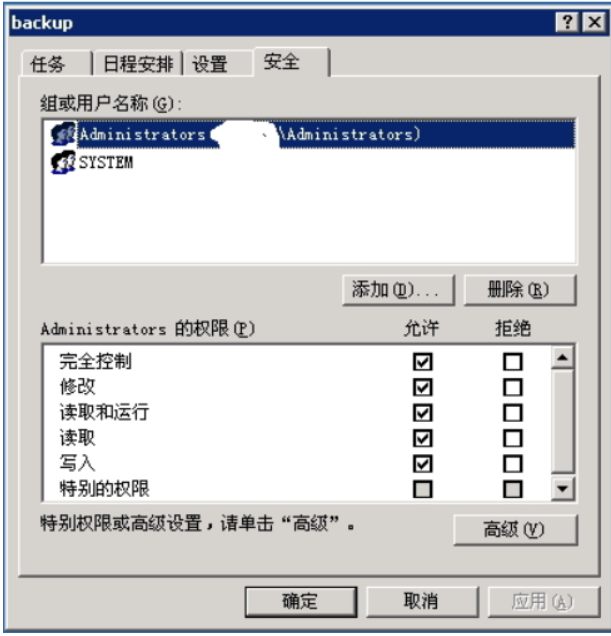
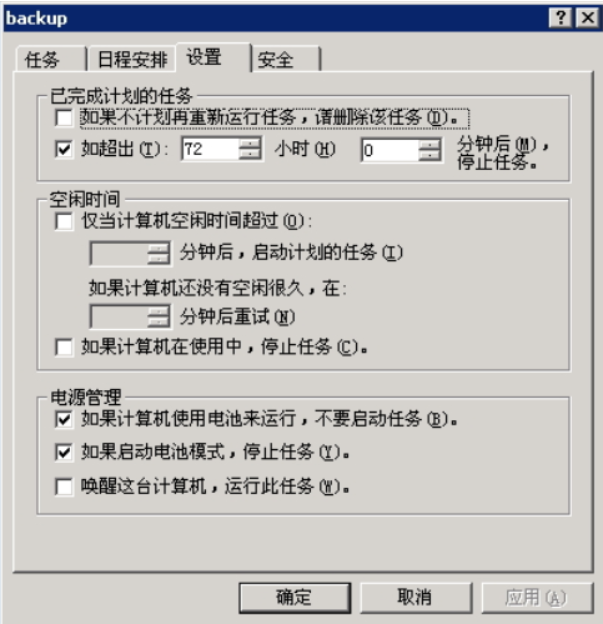
expdp '%USER%/%PASSWORD%@%DATABASE% as sysdba' directory=dump\_dir dumpfile=data\_%BACKUPDATE%.dmp full=y;

exit

6.6.2创建 windows任务计划

、





6.6.3编写拷贝程序

import java.io.\*;

import java.util.\*;

import java.net.URL;

import java.text.SimpleDateFormat;

public class BackupFile{

//1、获取当前路径

//2、读取当前路径下的属性文件，获取源文件夹和目标文件夹所在目录

//3、拷贝源文件夹下的所有内容至目标文件夹

//4、清空源文件夹

//5、保留30天以内的数据库备份

public static void main(String args[]){

//获取当前路径

String path = getCurrentPath();

File file = new File(path + "/dirIndex.properties" );

if(!file.exists()){

System.out.println("配置文件不存在！");

System.exit(1);

}

//读取当前路径下的属性文件，获取源文件夹和目标文件夹所在目录

Map<String,String> dirConfig = getCopyPath();

String source = dirConfig.get("sourceDir");

String dest = dirConfig.get("destinationDir");

File sourceDir = new File(source);//源文件夹

File destinationDir = new File(dest);//目标文件夹

if(!sourceDir.exists()){

System.out.println("源文件夹不存在！");

System.exit(1);

}

if(!destinationDir.exists()){

System.out.println("目标文件夹不存在！");

//清空源文件夹

deleteSourceFileChildren(source);

System.exit(1);

}

//拷贝源文件夹下的所有文件至目标文件夹

copyFiles(source,dest);

//清空源文件夹

deleteSourceFileChildren(source);

//保留30天以内的数据库备份

retainData(dest);

}

/\*\*

\* 获取当前路径

\*/

public static String getCurrentPath(){

String path = Thread.currentThread().getContextClassLoader().getResource("").getPath();

return path;

}

/\*\*

\* 读取当前路径下的属性文件，获取源文件夹和目标文件夹所在目录

\*/

public static Map<String,String> getCopyPath(){

Map<String, String> propMap = new HashMap<String, String>();

ClassLoader loader = Thread.currentThread().getContextClassLoader();

URL url = loader.getResource("dirIndex.properties");

InputStream in = null;

try {

in = url.openStream();

Properties props = new Properties();

props.load(in);

// 加载属性列表

Iterator<String> it = props.stringPropertyNames().iterator();

while (it.hasNext()) {

String key = it.next();

String value = props.getProperty(key);

propMap.put(key, value);

}

} catch (IOException ioe) {

ioe.printStackTrace();

}finally{

try {

in.close();

} catch (IOException e) {

e.printStackTrace();

}

}

return propMap;

}

/\*\*

\* 拷贝源文件夹下的所有内容至目标文件夹

\*/

public static void copyFiles(String sourceFile,String destinationFile){

Date date = new Date();

SimpleDateFormat sbf = new SimpleDateFormat("yyyyMMdd");

String fileName = sbf.format(date);

System.out.println(fileName);

String destFileStr = destinationFile + "/" + fileName;

File destFile = new File(destFileStr);

if(destFile.exists()){

deleteDir(destFile);//如果存在，删除该目录

}

try{

destFile.mkdirs();

File inputFile = new File(sourceFile);

File[] files = inputFile.listFiles();

FileInputStream input = null;

FileOutputStream output = null;

long start = System.currentTimeMillis();

copyFile(sourceFile,destFileStr);//拷贝所有内容至目标文件夹

long end = System.currentTimeMillis();

System.out.println("共耗时：" + (end - start) + "ms." );

}catch(Exception e){

e.printStackTrace();

}

}

//清空源文件夹

public static void deleteSourceFileChildren(String sourceFilePath){

File file = new File(sourceFilePath);

if(file.exists()){

deleteDir(file);

file.mkdirs();

}

}

//保留30天以内的数据库备份

public static void retainData(String dataPath){

File file = new File(dataPath);

File[] children = file.listFiles();

try{

Date date = new Date();

SimpleDateFormat sbf = new SimpleDateFormat("yyyyMMdd");

String dateStr = sbf.format(date);

String dirNames[] = new String[children.length];

for(int i=0; i<dirNames.length; i++){

File child = children[i];

if(child.isDirectory()){

dirNames[i] = child.getName();

}

}

System.out.println("文件夹长度：" + dirNames.length);

//如果备份数量小于30，则不删除

if(dirNames.length <= 30){

System.out.println("备份文件小于等于30份，不做删除。");

}else{

//如果备份数量大于30，则删除剩余的几个

List<Integer> dirNum = new ArrayList<Integer>();

for( String dirName : dirNames){

if(dirName.matches("[0-9]{8}")){

dirNum.add(Integer.parseInt(dirName));

}

}

Integer[] dirArr = new Integer[1];

dirArr = dirNum.toArray(dirArr);

Arrays.sort(dirArr);

dirArr = Arrays.copyOfRange(dirArr,0,dirArr.length - 30);

for(int i=0; i<dirArr.length; i++){

deleteDir(new File(dataPath + "/" + dirArr[i] + "" ) );

}

}

}catch(Exception e){

e.printStackTrace();

}

}

/\*\*

\* 拷贝文件夹下所有内容（文件夹和文件）到另一个文件夹

\*/

public static boolean copyFile(String sourceStr,String destStr){

File[] children = new File(sourceStr).listFiles();

FileInputStream input = null;

FileOutputStream output = null;

try{

for(int i=0; i<children.length; i++){

if(children[i].isDirectory()){

String newFilePath = destStr + "/" + children[i].getName();

File newFile = new File(newFilePath);

newFile.mkdir();

copyFile( (sourceStr + "/" + children[i].getName()),newFilePath);

}else{

input = new FileInputStream(sourceStr + "/" + children[i].getName() );

output = new FileOutputStream(destStr + "/" + children[i].getName() );

byte[] data = new byte[1024 \* 512];

int len;

while( (len = input.read(data)) != -1 ){

output.write(data,0,len);

}

input.close();

output.close();

}

}

}catch(Exception e){

e.printStackTrace();

return false;

}

return true;

}

/\*\*

\* 删除目录及目录下所有内容

\*/

public static boolean deleteDir(File file){

if(file.isDirectory()){

String[] children = file.list();

for(int i=0; i<children.length; i++){

boolean isSuccess = deleteDir(new File(file,children[i]));

if(!isSuccess){

return isSuccess;

}

}

}

return file.delete();

}

}

1 #源目录

2 sourceDir=E:/backup/backuptempdir

3 #目标目录

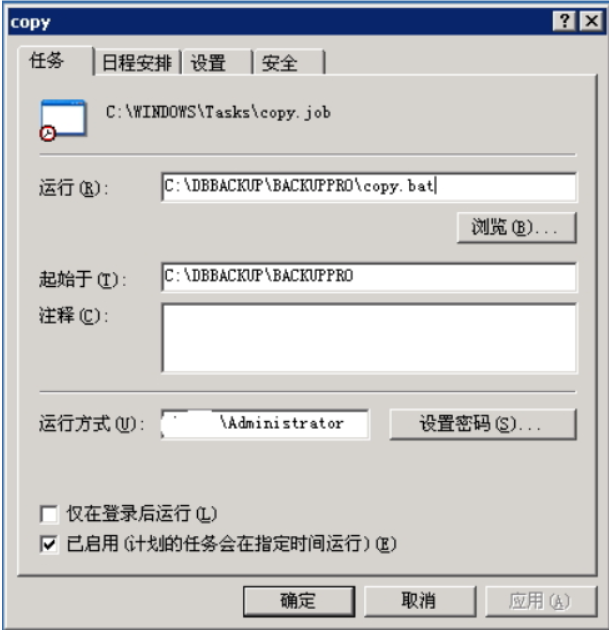
4 destinationDir=H:/

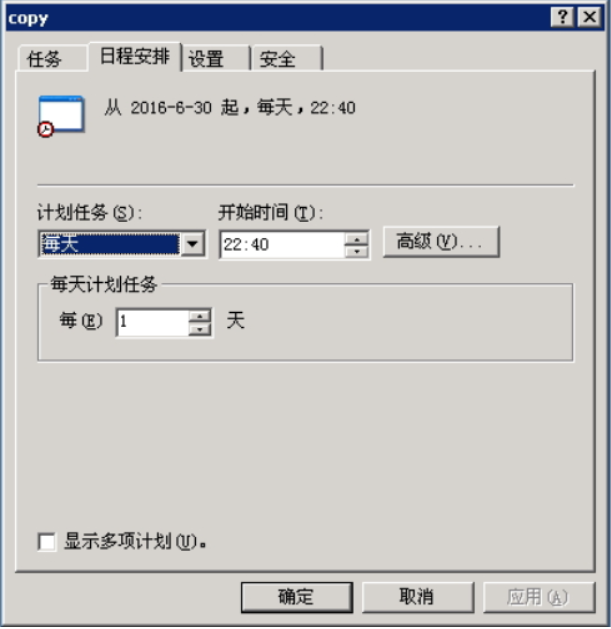
6.6.4 编写拷贝脚本

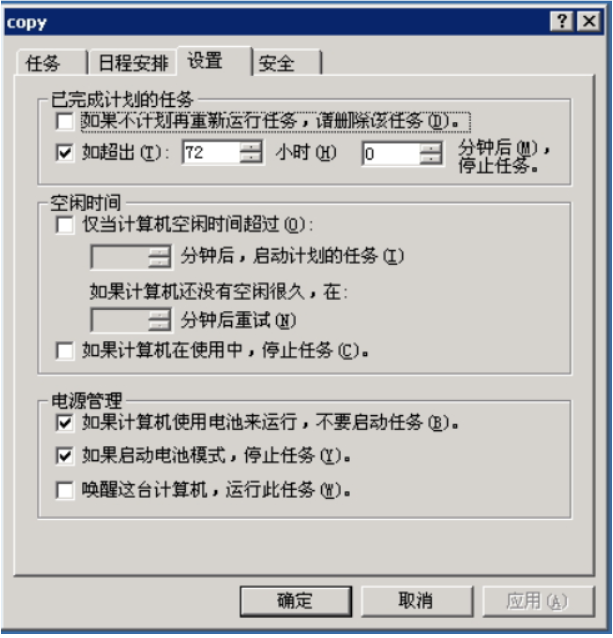
1 cd C:\DBBACKUP\BACKUPPRO

2 java BackupFile

6.6.5设定windows任务计划









# 7.项目总结

本次大作业是对这学期所学的综合时间，在这一次作业中，我设计了一个个人博客的数据库项目，运用到了表及表空间使用方案，涉及到了2个表空间6张表，插入了7万条数据，在权限设计的用户分配方案阶段，我设计了两类角色，游客和用户，用户有更高级的权限，而游客仅仅能够访问别人的文章，和评论的功能。在关于备份的设计上，我借鉴了网上自动备份的方案，通过编写脚本和进行设置，直接进行自动备份。

本次项目作为所有所学知识点的综合应用，对于我而言还是一个较大的挑战，在通过搜集相关资料的情况下一步步完成项目，完成这个项目也让我收益颇丰，感觉到了完成项目的成就感。