五、编程题

266、1 已知 Linux 系统为 CentOs7 系统,已知 hadoop 启动目录/simple/hadoop2.7.3/sbin,请依次写出命令。启动 hadoop、查看 hadoop 进程是否全部启动、递归查看当前 HDFS 有那些文件、在 HDFS 的根目录创建 input 目录、在本地/simple 目录下创建文件 data.txt 并上传到 HDFS 的/input 目录下、将集群上的/input/data.txt 下载到/appdata(该目录已存在)、最后关闭 hadoop

[root@master /]#cd /simple/hadoop2.7.3/sbin

[root@master sbin]#start-all.sh

[root@master sbin]#jps

[root@master sbin]#hadoop fs -ls -R /

[root@master sbin]#hadoop fs -mkdir /input

[root@master sbin]#cd /simple

[root@master simple]#touch data.txt

[root@master simple]#hadoop fs -put data.txt /input

[root@master simple]#hadoop fs -get /input/data.txt /appdata

[root@master simple]#cd /simple/hadoop2.7.3/sbin

[root@master sbin]#stop-all.sh

267、已知当前 hadoop 已全部正常启动，且 HDFS 的根目录下不存在 hdfstest 的目录，用到的 ip 和端口写 192.168.1.26:9000
使用 java api，在 HDFS 的根目录下，创建名为 hdfstest 的目录

import org.apache.hadoop.fs.FileSystem;

import java.net.URI;

import org.apache.hadoop.conf.Configuration;

```
import org.apache.hadoop.fs.Path;

import java.io.IOException;

public class MakeDir{

    public static void main(String[] args)throws IOException,InterruptedException{

        FileSystem fs = FileSystem.get(new URI("hdfs://192.168.1.26:9000"), new Configuration(), "root");

        Boolean flag = fs.mkdirs(new Path("/hdfstest"));

        System.out.println(flag?"创建成功":"创建失败");

    }

}
```

268、问题:编写一个程序，该程序接受控制台以逗号分隔的数字序列，并生成包含每个数字的列表和元组。假设向程序提供以下输入:

34 岁,67 年,55 岁,33 岁,12 日,98 年

则输出为:['34'， '67'， '55'， '33'， '12'， '98']

('34'， '67'， '55'， '33'， '12'， '98')

提示:在为问题提供输入数据的情况下，应该假设它是控制台输入。方法可以将列表转换为元组

解决方案:

```
import re

print('请输入一组数字：')

values=input()

l=values.split(",")

k=re.findall(r'[0-9]+',values)

t=tuple(k)

print (k)

print (t)
```

269、请在下面程序的下划线中补充完整程序（共 8 处）。

```
public class WordCount {
  publicstatic class TokenizerMapper extends
    Mapper<__Object____, _Text _ , __Text____, __IntWritable___> {
     private final static IntWritable one = newIntWritable(1);
     private Text word = new Text();
     public void map(LongWritable key, Text value,Context context){
        StringTokenizeritr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
          word.set(itr.nextToken());
          context.write(word, one);
        }
     }
  }
 public static class IntSumReducer extends
         Reducer< ___Text___ , ___IntWritable____ , Text, IntWritable> {
      private IntWritable result = newIntWritable();
      public void reduce( _Texy___ key, Iterable< __IntWritable___ > values, Context context)
{
          int sum = 0;
          for (IntWritable val : values) {
              sum += val.get();
          }
          result.set(sum);
          context.write(key, result);
      }
  }
 public static void main(String[] args) throws Exception {
    略.....
  }
}
```

270、编写 shell 脚本，能够生成 32 位随机密码

```
cat /dev/urandom | head -1 | md5sum | head -c 32
```

271、不退出数据库,完成备份 mingongge 数据库

```
 system mysqldump -uroot -pMgg123.0. -B mingongge >/root/mingongge_bak.sql
```

272、创建一 innodb GBK 表 test，字段 id int(4)和 name varchar(16)

```
create table test (
```

```
id int(4),

name varchar(16)

)ENGINE=innodb DEFAULT CHARSET=gbk;
```

273、利用条件运算符的嵌套来完成此题：学习成绩>=90 分的同学用 A 表示，60-89 分之间的用 B 表示，60 分以下的用 C 表示。

程序分析：(a>b)?a:b 这是条件运算符的基本例子。

```
public class Programme5 {

public static void main(String[] args) {

    System.out.println("请输入你的分数：");

    Scanner scanner=new Scanner(System.in);

     int input=scanner.nextInt();//获取输入

//等级判断

    String belong=input>=90?"A":(input>=60?"B":"c");


    System.out.println(input+"分属于："+belong);

    scanner.close();

}
```

274、计算字符串中子串出现的次数

```
public class Prog49{
public static void main(String[] args){
String str = "I come from County DingYuan Province AnHui.";
char[] ch = str.toCharArray();
int count = 0;
for(int i=0;i<ch.length;i++){
if(ch[i]==' ')
  count++;
}
```

```
count++;
System.out.println("共有"+count+"个字串");
}
}
```

275、809*??=800*??+9*??+1

其中??代表的两位数,8*??的结果为两位数，9*??的结果为 3 位数。求??代表的两位数，及809*??后的结果。

```
public class Prog42{
public static void main(String[] args){
int n = 0;
boolean flag = false;
for(int i=10;i<100;i++)
 if(809*i==800*i+9*i+1){
   flag = true;
   n = i;
   break;
 }
if(flag)
 System.out.println(n);
else
 System.out.println("无符合要求的数！");
}
}
```

276、海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子凭据分为五份，多了一个，这只猴子把多的一个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，又多了一个，它同样把多的一个扔入海中，拿走了一份，第三、第四、第五只猴子都是这样做的，问海滩上原来最少有多少个桃子？

```
public class Prog41{
public static void main(String[] args){
int n;
n = fun(0);
System.out.println("原来有"+n+"个桃子");
}
private static int fun(int i){
if(i==5)
  return 1;
```

```
else
  return fun(i+1)*5+1;
}
}
```

277、字符串排序。

```
public class Prog40{
public static void main(String[] args){
String[] str = {"abc","cad","m","fa","f"};
for(int i=str.length-1;i>=1;i--){
for(int j=0;j<=i-1;j++){
if(str[j].compareTo(str[j+1])<0){
String temp = str[j];
str[j] = str[j+1];
str[j+1] = temp;
}
}
}
for(String subStr:str)
  System.out.print(subStr+" ");
}
}
```

278、打印出杨辉三角形（要求打印出 10 行如下图）

程序分析：

```
          1

        1 1

      1 2 1

    1 3 3 1

  1 4 6 4 1

1 5 10 10 5 1
```

```
public class Prog33{
```

```
public static void main(String[] args){
int[][] n = new int[10][21];
n[0][10] = 1;
for(int i=1;i<10;i++)
  for(int j=10-i;j<10+i+1;j++)
    n[i][j] = n[i-1][j-1]+n[i-1][j+1];
for(int i=0;i<10;i++){
for(int j=0;j<21;j++){
if(n[i][j]==0)
  System.out.print("    ");
else{
    if(n[i][j]<10)
      System.out.print("   "+n[i][j]);//空格为了美观需要
    else if(n[i][j]<100)
      System.out.print(" "+n[i][j]);
      else
        System.out.print(n[i][j]);
  }
}
System.out.println();
}
}
}
```

279、取一个整数 a 从右端开始的 4～7 位。

程序分析：可以这样考虑：

(1)先使 a 右移 4 位。

(2)设置一个低 4 位全为 1,其余全为 0 的数。可用~(~0<<4)

(3)将上面二者进行&运算

```
import java.util.Scanner;
public class Prog32{
public static void main(String[] msg){
//输入一个长整数
Scanner scan = new Scanner(System.in);
long l = scan.nextLong();
```

```
scan.close();
//以下截取字符
String str = Long.toString(l);
char[] ch = str.toCharArray();
int n = ch.length;
if(n<7)
  System.out.println("输入的数小于 7 位！");
else
  System.out.println("截取的 4~7 位数字："+ch[n-7]+ch[n-6]+ch[n-5]+ch[n-4]);
}
}
```

280、求一个 3*3 矩阵对角线元素之和

程序分析：利用双重 for 循环控制输入二维数组，再将 a[i][i]累加后输出。

```
public class Prog29{
public static void main(String[] args){
int[][] a = new int[][] {{100,2,3,},{4,5,6},{17,8,9}};
matrSum(a);
}
private static void matrSum(int[][] a){
int sum1 = 0;
int sum2 = 0;
for(int i=0;i<a.length;i++)
  for(int j=0;j<a[i].length;j++){
    if(i==j) sum1 += a[i][j];
    if(j==a.length-i-1) sum2 += a[i][j];
  }
System.out.println("矩阵对角线之和分别是："+sum1+"和"+sum2);
}
}
```

281、给一个不多于 5 位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。

```
public class Prog24{
public static void main(String[] args){
int n = Integer.parseInt(args[0]);
int i = 0;
int[] a = new int[5];
```

```
do{
a[i] = n%10;
  n /= 10;
  ++i;
}while(n!=0);
System.out.print("这是一个"+i+"位数，从个位起，各位数字依次为：");
for(int j=0;j<i;j++)
  System.out.print(a[j]+" ");
}
}
```

290、有 5 个人坐在一起，问第五个人多少岁？他说比第 4 个人大 2 岁。问第 4 个人岁数，他说比第 3 个人大 2 岁。问第三个人，又说比第 2 人大两岁。问第 2 个人，说比第一个人大两岁。最后问第一个人，他说是 10 岁。请问第五个人多大？

程序分析：利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人岁数，需知道第四人的岁数，依次类推，推到第一人（10 岁），再往回推。

```
public class Prog23{
public static void main(String[] args){
System.out.println(getAge(5,2));
}
//求第 m 位同志的年龄
private static int getAge(int m,int n){
if(m==1)
  return 10;
else
  return getAge(m-1,n)+n;
}
}
```

291、利用递归方法求 5!。

程序分析：递归公式：fn=fn_1*4!

```
public class Prog22{
public static void main(String[] args){
System.out.println(fact(10));
}
//递归求阶乘
```

```
private static long fact(int n){
if(n==1)
  return 1;
else
  return fact(n-1)*n;
}
}
```

292、求 1+2!+3!+...+20!的和

程序分析：此程序只是把累加变成了累乘。

```
public class Prog21{
public static void main(String[] args){
long sum = 0;
for(int i=0;i<20;i++)
  sum += factorial(i+1);
System.out.println(sum);
}
//阶乘
private static long factorial(int n){
int mult = 1;
for(int i=1;i<n+1;i++)
  mult *= i;
return mult;
}
}
```

293、打印出如下图案（菱形）

```
   *

  ***

 *****

*******

 *****

  ***

   *
```

```
public class Prog19{
public static void main(String[] args){
int n = 5;
printStar(n);
}
//打印星星
private static void printStar(int n){
//打印上半部分
for(int i=0;i<n;i++){
for(int j=0;j<2*n;j++){
   if(j<n-i)
     System.out.print(" ");
   if(j>=n-i && j<=n+i)
     System.out.print("*");
  }
  System.out.println();
}
//打印下半部分
for(int i=1;i<n;i++){
System.out.print(" ");
for(int j=0;j<2*n-i;j++){
if(j<i)
     System.out.print(" ");
   if(j>=i && j<2*n-i-1)
     System.out.print("*");
}
System.out.println();
}
}
}
```

294、有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

```
public class Prog11{
public static void main(String[] args){
int count = 0;
int n = 0;
for(int i=1;i<5;i++){
for(int j=1;j<5;j++){
if(j==i)
  continue;
for(int k=1;k<5;k++){
```

```
if(k!=i && k!=j){
n = i*100+j*10+k;
  System.out.print(n+" ");
  if((++count)%5==0)
  System.out.println();
}
}
}
}
System.out.println();
System.out.println("符合条件的数共："+count+"个");
}
}
```

295、输入一个字符，判断它是否为小写字母，如果是，将它转换成大写字母，否则，不转换。

```
package HomeWork03;

import java.util.Scanner;

public class HomeWork03 {

    public static void main(String[] args) {

        //小写字母的 ascll 值为 97-122

        //大写字母的 ascll 值为 65-90

        System.out.println("请输入一个字母：\n");

        Scanner input = new Scanner(System.in);

        char zimu=input.next().charAt(0);

          if (zimu>=97&&zimu<=122){          //判断是否是小写字母

              System.err.println("该字母是小写字母");

              zimu=(char) (zimu-32);       //如果是小写字母则 将其转换成大写字母

              System.err.println("转换之后的大写字母是："+zimu);
```

```
        }
        else{
            System.out.println("该字母不是小写字母！");
        }
    }
}
```

295、用 while 循环，计算 1~200 之间所有 3 的倍数之和

```
package HomeWork09;
public class HomeWork09 {
    public static void main(String[] args) {
        // 用 while 循环，计算 1~200 之间所有 3 的倍数之和。
        int a=1;
        int sum=0;
        while(a<=200){
            if(a%3==0){
                sum=sum+a;
            }
            a++;
        }
        System.out.println("1~200 之间所有 3 的倍数之和为:"+sum);
    }
}
```

296、编写程序，输出 200~500 之间的所有素数。

```
package HomeWork10;

public class HomeWork10 {

    public static void main(String[] args) {

        int num=200;

        while (num<=500) {

            boolean tag=true;       //素数标记

            for(int d=2;d<=num-1;d++){

                if(num % d==0){

                    tag=false;

                    break;

                }

            }

            if(tag){                //如果是素数

                System.out.println(num);

            }

            num++;

        }

    }

}
```

297、使用循环语句输出下面的图形。

```
#
# # #
```

#####

#######

#########

```
package HomeWork12;
public class HomeWork12 {
    public static void main(String[] args) {
        int aa=-1;
        for( int a=0;a<5;a++){
            aa+=2;
            for(int b=1;b<=aa;b++){
                System.out.print( "#"  );
            }
            System.out.println();}

    }
}
```

298、输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

1.程序分析：利用 while 语句,条件为输入的字符不为'\n'.

```
package cn.edu.hit;

import java.util.Scanner;

public class strIdentify {

    public static void main(String[] args) {

        int abcCount = 0;

        int spaceCount = 0;

        int numCount = 0;

        int otherCount = 0;

        Scanner sc = new Scanner(System.in);

        String str = sc.nextLine();
```

```
            char[] ch = str.toCharArray();
            for (int i = 0; i < ch.length; i++) {
                if (Character.isDigit(ch[i])) {
                    numCount++;
                } else if (Character.isSpaceChar(ch[i])) {
                    spaceCount++;
                } else if (Character.isLetter(ch[i])) {
                    abcCount++;
                } else {
                    otherCount++;
                }
            }
            System.out.println("字母个数"+abcCount);
            System.out.println("数字个数"+numCount);
            System.out.println("空格个数"+spaceCount);
            System.out.println("其他字符个数"+otherCount);
    }
}
```

299、一个数如果恰好等于它的因子之和，这个数就称为"完数"。例如 6=1 + 2 + 3. 编程 找出 1000 以内的所有完数。

```
public class wanShu {
    public static void main(String[] args) {
        int k = 2;
```

```
        int num = 0;
        int temp = 1;
        int j = 0;
        for (num = 1; num <= 1000; num++) {
            k = 2;
            temp = 1;
            j = num;
            while (j >= k) {
                if (j % k == 0) {
                    temp += k;
                    j = j / k;
                } else {
                    k++;
                }
            }
            if (temp == num) {
                System.out.println(temp);
            }

        }
    }
}
```

300、一球从 100 米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第 10 次落地时，共经过多少米？第 10 次反弹多高？

```
public class testBall {
    public static void main(String[] args) {
        double a = 100;
        double sum = 100;
        for(int i =2 ;i<=10;i++){
            a = a*0.5;
            sum += a*2;
        }
        System.out.println("a="+a);
        System.out.println("sum="+sum);
    }
}
```

实操题题库

第一题：Sqoop--Hadoop 和关系型数据库中的数据相互转移的工具

2.2.1 下载并解压

1）下载地址：http://mirrors.hust.edu.cn/apache/sqoop/1.4.6/

2）上传安装包 sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz 到 hadoop102 的/opt/software 路径中

3）解压 sqoop 安装包到指定目录，如：

[root@hadoop102 software]$ tar -zxf sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz -C /opt/module/

4）解压 sqoop 安装包到指定目录，如：

[root@hadoop102 module]$ mv sqoop-1.4.6.bin__hadoop-2.0.4-alpha/ sqoop

2.2.2 修改配置文件

1）进入到/opt/module/sqoop/conf 目录，重命名配置文件

[root@hadoop102 conf]$ mv sqoop-env-template.sh sqoop-env.sh

2）修改配置文件

[root@hadoop102 conf]$ vim sqoop-env.sh

增加如下内容

export HADOOP_COMMON_HOME=/opt/module/hadoop-3.1.3

export HADOOP_MAPRED_HOME=/opt/module/hadoop-3.1.3

export HIVE_HOME=/opt/module/hive

export ZOOKEEPER_HOME=/opt/module/zookeeper-3.5.7

export ZOOCFGDIR=/opt/module/zookeeper-3.5.7/conf

2.2.3 拷贝 JDBC 驱动

1）将 mysql-connector-java-5.1.48.jar 上传到/opt/software 路径

2）进入到/opt/software/路径，拷贝 jdbc 驱动到 sqoop 的 lib 目录下。

[root@hadoop102 software]$ cp mysql-connector-java-5.1.48.jar /opt/module/sqoop/lib/

2.2.4 验证 Sqoop

我们可以通过某一个 command 来验证 sqoop 配置是否正确：

[root@hadoop102 sqoop]$ bin/sqoop help

出现一些 Warning 警告（警告信息已省略），并伴随着帮助命令的输出：

Available commands:

```
  codegen            Generate code to interact with database records
  create-hive-table     Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables     Import tables from a database to HDFS
  import-mainframe    Import datasets from a mainframe server to HDFS
  job                Work with saved jobs
  list-databases        List available databases on a server
  list-tables           List available tables in a database
  merge              Merge results of incremental imports
  metastore           Run a standalone Sqoop metastore
  version            Display version information
```

2.2.5 测试 Sqoop 是否能够成功连接数据库

[root@hadoop102 sqoop]$ bin/sqoop list-databases --connect jdbc:mysql://hadoop102:3306/ --username root --password 000000

出现如下输出：

information_schema

metastore

mysql

oozie

performance_schema

sqoop

第二题：Liun 下 MySQL 安装

安装包准备

1）卸载自带的 Mysql-libs（如果之前安装过 mysql，要全都卸载掉）

[root@hadoop102 software]$ rpm -qa | grep -i -E mysql\|mariadb | xargs -n1 sudo rpm -e --nodeps

2）将安装包和 JDBC 驱动上传到/opt/software，共计 6 个

01_mysql-community-common-5.7.29-1.el7.x86_64.rpm

02_mysql-community-libs-5.7.29-1.el7.x86_64.rpm

03_mysql-community-libs-compat-5.7.29-1.el7.x86_64.rpm

04_mysql-community-client-5.7.29-1.el7.x86_64.rpm

05_mysql-community-server-5.7.29-1.el7.x86_64.rpm

mysql-connector-java-5.1.48.jar

2.1.2 安装 MySQL

1）安装 mysql 依赖

[root@hadoop102 software]$ sudo rpm -ivh 01_mysql-community-common-5.7.29-1.el7.x86_64.rpm

[root@hadoop102 software]$ sudo rpm -ivh 02_mysql-community-libs-5.7.29-1.el7.x86_64.rpm

[root@hadoop102 software]$ sudo rpm -ivh 03_mysql-community-libs-compat-5.7.29-1.el7.x86_64.rpm

2）安装 mysql-client

[root@hadoop102 software]$ sudo rpm -ivh 04_mysql-community-client-5.7.29-1.el7.x86_64.rpm

3）安装 mysql-server

[root@hadoop102 software]$ sudo rpm -ivh 05_mysql-community-server-5.7.29-1.el7.x86_64.rpm

4）启动 mysql

[root@hadoop102 software]$ sudo systemctl start mysqld

5）查看 mysql 密码

[root@hadoop102 software]$ sudo cat /var/log/mysqld.log | grep password

2.1.3 配置 MySQL

配置只要是 root 用户+密码，在任何主机上都能登录 MySQL 数据库。

1）用刚刚查到的密码进入 mysql（如果报错，给密码加单引号）

[root@hadoop102 software]$ mysql -uroot -p’ password’

2）设置复杂密码(由于 mysql 密码策略，此密码必须足够复杂)

mysql> set password=password("Qs23=zs32");

3）更改 mysql 密码策略

mysql> set global validate_password_length=4;

mysql> set global validate_password_policy=0;

4）设置简单好记的密码

mysql> set password=password("000000");

5）进入 msyql 库

mysql> use mysql

6）查询 user 表

mysql> select user, host from user;

7）修改 user 表，把 Host 表内容修改为%

mysql> update user set host="%" where user="root";

8）刷新

mysql> flush privileges;

9）退出

mysql> quit;

第三题：mongodb 数据库相关（20 分）

show dbs:显示数据库列表

show collections：显示当前数据库中的集合（类似关系数据库中的表 table）

show users：显示所有用户

use yourDB：切换当前数据库至 yourDB

db.help() ：显示数据库操作命令

db.yourCollection.help() ：显示集合操作命令，yourCollection 是集合名

MongoDB 没有创建数据库的命令，如果你想创建一个“School”的数据库，先运行 use School 命令，之后做一些操作

1.创建一个“student”的数据库

use student

2.插入数据

db.student.insert({_id:1, sname: ‘zhangsan’, sage: 20})

或者

db.student.save({_id:1, sname: ‘zhangsan’, sage: 22})

这两种方式，其插入的数据中_id 字段均可不写，会自动生成一个唯一的_id 来标识本条数据。而 insert 和 save 不同之处在于：在手动插入_id 字段时，如果_id 已经存在，insert 不做操作，save 做更新操作；如果不加_id 字段，两者作用相同都是插入数据。

3.查找数据

db.student.find(criteria, filterDisplay)

criteria ：查询条件，可选

filterDisplay：筛选显示部分数据，如显示指定列数据，可选（当选择时，第一个参数不可省略，若查询条件为空，可用{}做占位符，如下例第三句）

db.student.find() #查询所有记录。相当于：select * from student

db.student.find({sname: ‘lisi’}) #查询 sname=’lisi’的记录。相当于： select＊from student where sname=‘lisi’

db.student.find({},{sname:1, sage:1}) #查询指定列 sname、sage 数据。相当于：select sname,sage from student。sname:1 表示返回 sname 列，默认_id 字段也是返回的，可以添加_id:0（意为不返回_id）写成{sname: 1, sage: 1,_id:0}，就不会返回默认的_id 字段了

db.student.find({sname: ‘zhangsan’, sage: 22}) #and 与条件查询。相当于：select * from student where sname = ‘zhangsan’ and sage = 22

db.student.find({$or:[{sage:22}, {sage:25}]}) #or 条件查询。相当于：select＊from student where sage = 22 or sage = 25

4.修改数据

db.student.update({sname: ‘lisi’}, {$set: {sage: 30}}, false, true) #相当于：update student set sage =30 where sname = ‘lisi’;

5.删除数据

db.student.remove({sname: ‘zhaoliu’}) #相当于：delete from student where sname=‘zhaoliu’

6.删除集合

db.student.drop()

第四题： 设计题（20 分）

在一个教师信息管理系统中提供以下信息：

系：系代号、系名、联系电话、联系地址。

教师：教师号、姓名、性别、职称。

课程：课程代号、课程名、课程简介。

学科方向：学科代码、学科名称、研究内容。

在一个教师信息管理系统中提供以下信息：

系：系代号、系名、联系电话、联系地址。

教师：教师号、姓名、性别、职称。

课程：课程代号、课程名、课程简介。

学科方向：学科代码、学科名称、研究内容。

上述实体集中存在如下联系：

（1）每个系有唯一代号，但可能有多个联系电话；

（2）每个系可能拥有一个或者多个学科方向，某一学科方向只能属于某一固定系；

（3）每个系可能拥有多名教师，但至少拥有一名教师，一名教师只能属于某一固定系；

（4）教师只能属于某一学科方向，但允许某些教师暂时不属于任何学科方向，也允许某一学科方向暂时没有教师；

（5）一个教师可以讲授多门课程，一门课程可为多个教师讲授，教师只在某一固定学期讲授课程，学期用学期号表示，允许教师暂时不讲授课程；

（6）每个学科方向只能有一名教师作为学科带头人，也可能暂时空缺；

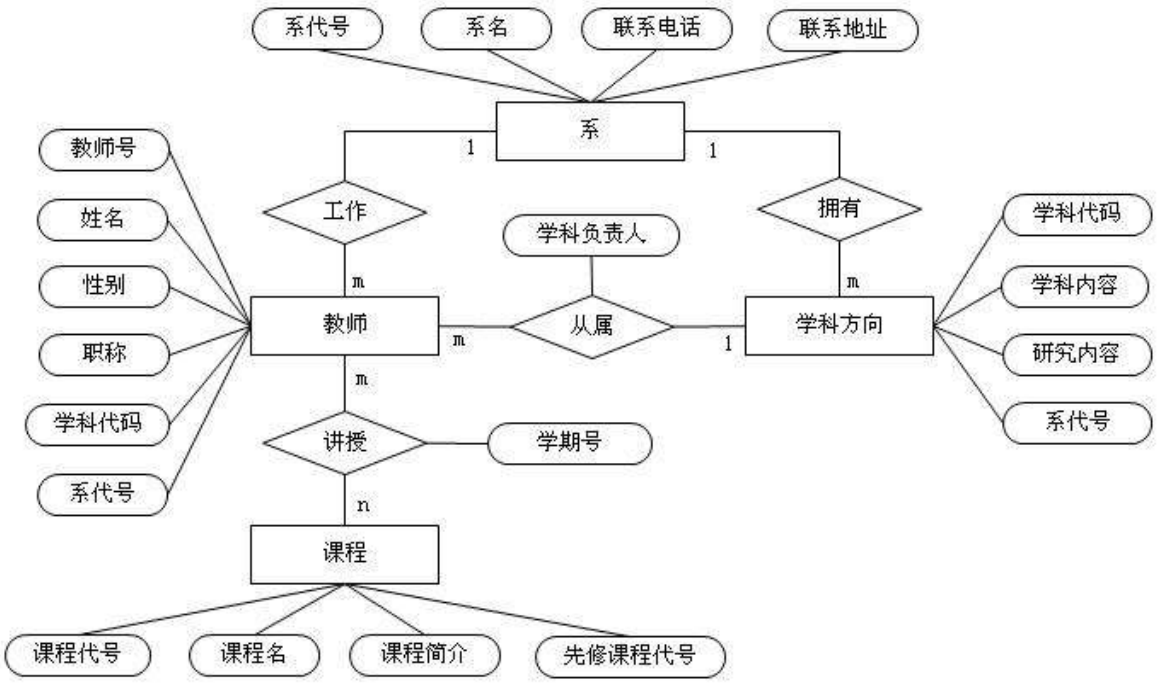
（7）某些课程有可能有先修课程，但最多只能有一门先修课程，一门课程可能为多门课程的先修课程。

试完成如下设计：

（1）构造满足需求的 E/R 图模型，并对模型图进行必要文档描述，必要时可以对需求做合理补充说明，但在文档中应该描述清楚。

（2）将第一步得到的 E/R 模型一步一步转换为等价的关系模式，要写出转换步骤和说明。

答：（1）根据题意，得 E-R 图如下：



E-R 图描述：

1、根据题意，在上述 E-R 图中有系、学科方向、教师、课程四个实体型。

2、某个学科方向只能属于某一固定系，可以将系作为学科方向的属性，在系实体型中，系代号可以作为唯一区分系的属性，因此补充系代号为学科方向的属性。同理，可将学科代码和系代号作为教师的属性，先修课程代号作为课程的属性。故上述 E-R 图的四个实体型的属性可以作以下描述：

1）系的属性有：系代号、系名、联系电话、联系地址；

2）学科方向的属性有：学科代码、学科内容、研究内容、系代号；

3）教师的属性有：教师号、姓名、性别、职称、学科代码、系代号；

4）课程的属性有：课程代号、课程名、课程简介、先修课程代号。

3、四个实体型之间的联系为：

1）系拥有一个或多个学科方向，某一学科方向只能属于某一固定系，系与学科方向为一对多的联系；

2）系拥有一名或多名教师，某一教师只能属于某一固定系，系与教师之间为一对多的联系；

3）教师只能属于某一学科方向，教师与学科方向为一对多的联系。

4）一个教师可以讲授多门课程，一门课程可为多名教师讲授，教师与课程之间为多对多的联系。

4、因为教师只在某一固定学期讲授课程，学期用学期号表示，故学期号可以作为讲授的属性，同理可得学科负责人为从属的属性。

（2）根据(1)的描述，E-R 图中实体型转换为等价的关系模式有：

系（系代号，系名，联系电话，联系地址） 主键：系代号

学科方向（学科代码，学科内容，研究内容，系代号） 主键：学科代码

教师（教师号，姓名，性别，职称，学科代码，系代号） 主键：教师号

课程（课程代号，课程名，课程简介，先修课程代号） 主键：课程代号

同理，根据(1)的描述，E-R 图中联系转换为等价的关系模式有：

从属（教师号，学科代码，学科负责人） 主键：教师号

讲授（教师号，课程代码，学期号） 主键：教师号+课程代码

联系转换为等价的关系模式与 M 端合并，得 E-R 图中转换为等价的关系模式有：

系（系代号，系名，联系电话，联系地址） 主键：系代号

学科方向（学科代码，学科内容，研究内容，系代号） 主键：学科代码

教师（教师号，姓名，性别，职称，学科代码，系代号，学科负责人否） 主键：教师号

课程（课程代号，课程名，课程简介，先修课程代号） 主键：课程代号

讲授（教师号，课程代码，学期号） 主键：教师号+课程代码

第五题： 备份服务器配置 rsync 文件 vi /etc/rsyncd.conf

#工作中指定用户(可以不指定为 0)

uid = 0

gid = 0

#相当于黑洞.出错定位

```
use chroot = no

#有多少个客户端同时传文件

max connections = 200

#超时时间

timeout = 300

#进程号文件

pid file = /var/run/rsyncd.pid

#日志文件

lock file = /var/run/rsync.lock

#日志文件

log file = /var/log/rsyncd.log

#模块开始

#模块名称随便起（可以是多个）

[backup]

#需要备份的目录

path = /backup

#表示出现错误忽略错误

ignore errors

#表示网络权限可写(本地控制真正可写)

read only = false

#这里设置 IP 或让不让同步

list = false

#指定允许的网段

hosts allow = 192.168.1.0/24

#拒绝链接的地址，一下表示没有拒绝的链接。
```

hosts deny = 0.0.0.0/32

#不要动的东西(默认情况)

#虚拟用户

auth users = rsync_backup

#虚拟用户的密码文件

secrets file = /etc/rsync.password

第六题：hdfs（20 分）

1.hdfs

有三种 shell 命令方式。

hadoop fs

hadoop dfs(已过期)

hdfs dfs（常用）

hadoop fs 适用于任何不同的文件系统，比如本地文件系统和 HDFS 文件系统

hadoop dfs 只能适用于 HDFS 文件系统

hdfs dfs 跟 hadoop dfs 的命令作用一样，也只能适用于 HDFS 文件系统

1.在 hdfs 创建一个在 user 下的目录 hadoop

hadoop fs -mkdir /user/hadoop

2.写“Hello world”到 yue.txt 文件

echo “Hello world”->yue.txt

3.将 test.txt 放入 hdfs 在 user 下的目录 hadoop

hadoop fs -put test.txt /user/hadoop

4.在 hdfs 中查看 test.txt

hadoop fs -cat /user/test.txt

5.从 hdfs 上下载 test.txt 到本地

hadoop fs -get /user/test.txt

6.从 hdfs 上删除 test.txt

hadoop fs -rm /user/test.txt

第七题：mongodb 数据库相关（20 分）

show dbs:显示数据库列表

show collections：显示当前数据库中的集合（类似关系数据库中的表 table）

show users：显示所有用户

use yourDB：切换当前数据库至 yourDB

db.help() ：显示数据库操作命令

db.yourCollection.help() ：显示集合操作命令，yourCollection 是集合名

MongoDB 没有创建数据库的命令，如果你想创建一个“School”的数据库，先运行 use School 命令，之后做一些操作

1.创建一个“student”的数据库

use student

2.插入数据

db.student.insert({_id:1, sname: ‘zhangsan’, sage: 20})

或者

db.student.save({_id:1, sname: ‘zhangsan’, sage: 22})

这两种方式，其插入的数据中_id 字段均可不写，会自动生成一个唯一的_id 来标识本条数据。而 insert 和 save 不同之处在于：在手动插入_id 字段时，如果_id 已经存在，insert 不做操作，save 做更新操作；如果不加_id 字段，两者作用相同都是插入数据。

3.查找数据

db.student.find(criteria, filterDisplay)

criteria ：查询条件，可选

filterDisplay：筛选显示部分数据，如显示指定列数据，可选（当选择时，第一个参数不可省略，若查询条件为空，可用{}做占位符，如下例第三句）

db.student.find() #查询所有记录。相当于：select * from student

db.student.find({sname: ‘lisi’}) #查询 sname=’lisi’的记录。相当于： select＊from student where sname= ‘lisi’

db.student.find({},{sname:1, sage:1}) #查询指定列 sname、sage 数据。相当于：select sname,sage from student。sname:1 表示返回 sname 列，默认_id 字段也是返回的，可以添加_id:0（意为不返回_id）写成{sname: 1, sage: 1,_id:0}，就不会返回默认的_id 字段了

db.student.find({sname: ‘zhangsan’, sage: 22}) #and 与条件查询。相当于：select * from student where sname = ‘zhangsan’ and sage = 22

db.student.find({$or:[{sage:22}, {sage:25}]}) #or 条件查询。相当于：select＊from student where sage = 22 or sage = 25

4.修改数据

db.student.update({sname: ‘lisi’}, {$set: {sage: 30}}, false, true) #相当于：update student set sage =30 where sname = ‘lisi’;

5.删除数据

db.student.remove({sname: ‘zhaoliu’}) #相当于：delete from student where sname= ‘zhaoliu’

6.删除集合

db.student.drop()

第八题：HDFS （20 分）

1.创建一个 student 表，属性有：name,sex,age,dept,course。

create ‘student’,‘name’,‘sex’,‘age’,‘dept’,‘course’

2.为 student 表添加了学号为 18001，名字为 zhangsan 的一行数据，其行键为 18001。

put ‘student’,’ 18001’,’ name’,’ zhangsan’

3.查看数据

get ‘student’,‘18001’

或者

scan ‘student’

4.删除 student 表中 95001 行下的 sex 列的所有数据。

delete ‘student’,‘18001’,‘sex’

5.删除 student 表中的 18001 行的全部数据。

deleteall ‘student’,‘18001’

6.删除表

disable ‘student’

drop ‘student’

第九题：Linuex 基本操作（20 分）

1.在当前目录下建立文件 exam.c，将文件 exam.c 拷贝到/tmp 这个目录下，并改名为 shiyan.c。

```
touch exam.c
cp /root/exam.c /tmp/shiyan.c
```

2. 在任何目录下回到用户主目录。

```
cd /tmp
cd
```

3. 打印当前目录（隐藏文件也显示）。

```
ll -a
```

4.在当前目录中新建文件 text 并设置文件的属性为文件属主(u)增加执行权限与文件属主同组用户(g)增加写权限其他用户(o) 删除读权限。

```
touch text
chmod u+x text
chmod g+w text
chmod o-r text
```

5. 创建用户 xu 和 liu 并将/home/xu 目录中的所有文件拷贝到目录/home/liu 中。

```
useradd xu
useradd liu
ll /home/xu
cd /home/xu
touch a1.c
cp -r /home/xu/* /home/liu
```

第十题：intellij idea 实操 （20 分）

生成并运行应用程序

完成“YUE,I LOVE YOU”的输出

将应用程序打包到 JAR 中

运行打包的应用程序

为打包的应用程序创建运行配置