# 数据库原理及其应用第二次小组作业

## 一、小组成员信息

**小组组号：**2

**小组成员：**税家晖　　41934063
刘彦升　　41934049
赵宏泽　　41933008

## 二、成语接龙结果

**（一）查找初始词为“不忘初心”的成语接龙序列，接龙条件首尾同字。**

| step | cy_res |
|---|---|
| 0 | 不忘初心 |
| 1 | 心安理得 |
| 2 | 得不偿失 |
| 3 | 失败为成功之母 |
| 4 | 母以子贵 |
| 5 | 贵耳贱目 |
| 6 | 目不识丁 |
| 7 | 丁公凿井 |
| 8 | 井臼亲操 |
| 9 | 操刀必割 |
| 10 | 割臂盟公 |

**（二） 查找初始词为“不忘初心”的成语接龙序列，接龙条件首尾同音。**

| step | cy_res |
|---|---|
| 0 | 不忘初心 |
| 1 | 心安理得 |
| 2 | 德薄才疏 |
| 3 | 数白论黄 |
| 4 | 慌不择路 |
| 5 | 路不拾遗 |
| 6 | 一把死拿 |
| 7 | 拿班作势 |
| 8 | 失败为成功之母 |
| 9 | 木本水源 |
| 10 | 援鳖失龟 |

## （三） 查找初始词为“不忘初心”，结束词为“牢记使命”的成语接龙序列，接龙条件首尾同音。

1)使用不限制成语序列长度的游标搜索算法，我们得到第一个，长度为 393 的可行成语序列：

| step | cy_res |
|---|---|
| 0 | 不忘初心 |
| 1 | 心安理得 |
| 2 | 德薄才疏 |
| 3 | 数白论黄 |
| 4 | 慌不择路 |
| 5 | 路不拾遗 |
| … | …… |
| 195 | 开诚布公 |
| 196 | 公报私仇 |
| 197 | 臭不可当 |
| 198 | 当耳边风 |
| 199 | 风不鸣条 |
| … | …… |
| 389 | 炙冰使燥 |
| 390 | 皂白不分 |
| 391 | 分崩离析 |
| 392 | 喜从天降 |
| 393 | 江东父老 |

2)使用从初始词与结束词同时出发的方法，我们得到了最短的可行成语序列为 4：

| pair | cy | cy |
|---|---|---|
| 1 | 信而好古 | 顾犬补牢 |
| 2 | 信马由缰 | 江东父老 |
| 3 | 心弛神往 | 往返徒劳 |
| 4 | 心弛神往 | 亡羊补牢 |
| 5 | 心病还须心药医 | 一举手之劳 |
| 5 | 心病还须心药医 | 倚老卖老 |
| 6 | 心病还须心药医 | 以逸待劳 |
| 7 | 心口如一 | 一举手之劳 |
| 8 | 心口如一 | 倚老卖老 |
| 9 | 心口如一 | 以逸待劳 |
| 10 | 心旷神怡 | 一举手之劳 |
| 11 | 心旷神怡 | 倚老卖老 |
| 12 | 心旷神怡 | 以逸待劳 |

## 三、过程中遇到的问题及解决

首先，在解决第一问的时候，中文字符的占位问题，及空格，导致了不能直接利用 substring 或者 right 等函数来取成语中的第一个字或者最后一个字，应当利用 patindex 函数找到“第一次出现汉字”的位置，然后再取出。

例如

```
@ch = substring(reverse(@str),patindex('%[吖-咗]%',reverse(@str)),1)
```

便可取出

第二个问题是，当发现某一个词无后继词怎么办？例如蛙鸣蝉噪，在整个成语库中我们无法找到以噪开头的成语。我们利用游标来解决这个问题。例如：

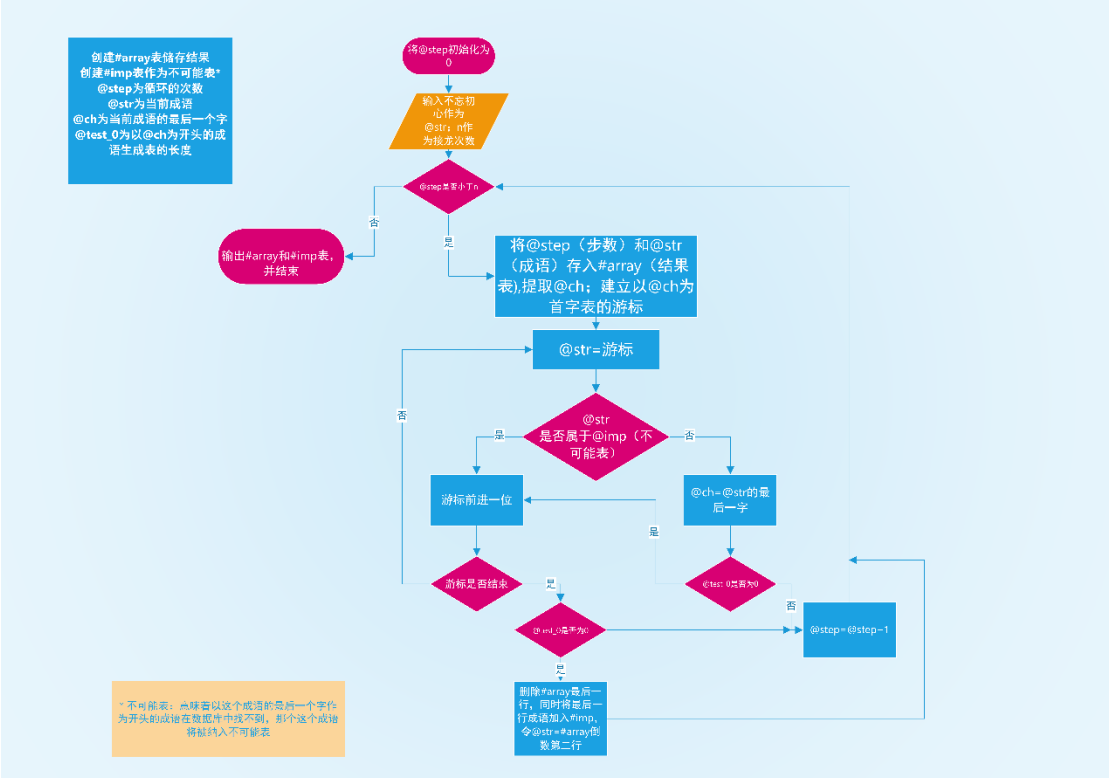
*井底之蛙→蛙鸣蝉噪*

我们对以“蛙”字开头的成语建立游标，如果发现蛙鸣蝉噪无后继词，则选择下一个以蛙字开头的成语加入结果表。

第三个问题是，如果发现井底之蛙后面只能接蛙鸣蝉噪怎么办？也就是说没有办法更换另一个后继词。这时我们就只能选择更换井底之蛙这个词，也就是倒退到丁公凿井继续进行接龙。

第二问和第三问就比较简单了，只需要将查询条件改为拼音查询即可。但拼音查询仍然存在空格字符，有些成语没有拼音的问题，我们在第四部分将展示如何解决该问题。

## 四、流程图及主要程序解释



## （一）程序一

首先我们先定义要用到的变量及临时表

```
declare @str nvarchar(50)='不忘初心'   --当前成语
declare @ch nvarchar(1)='心'           --当前成语末尾词
declare @step int = 0,@n int = 11      --Step表示当前层数，@n表示总层数，即接龙次数
declare @test_1 int, @test_2 int, @test_3 int
                                       --@test_1表示当前成语末尾词在成语表中检索结果数=0
表示无法接龙
                                       --@test_2表示测试@str是否已经在结果中出现过
                                          --@test_3表示测试@str是否出现在#imp不可能表
if object_id('tempdb.dbo.#array') is not null  --判断临时表是否已经存在
   drop table #array
create table #array(step int primary key, cy_res nvarchar(50), num int)--创建临时表，用
于显示接龙结果
if object_id('tempdb.dbo.#imp') is not null  --判断临时表是否已经存在
   drop table #imp
create table #imp(cy_imp nvarchar(50) primary key)--创建不可能进行下去的词表
```

其中#array 为结果存储表，#imp 表示不可能表，即在该表中的词已无法继续接龙。

```
while @step<@n                         --开始循环
     begin
     end
```

在循环体内，我们分为两个小块，第一块为判断变量@test_1 是否为 0，即在游标迭代完毕后，发现仍然不存在能够继续接龙下去的词，这边说明上一步有问题，我们要将上一步存储的词语添加到不可能表，并删除其在结果表中的记录，然后倒回到两步前的词。代码如下所示

```
if @test_1 = 0 or @str is NULL     --如果@test_1仍然为0，说明上一个词有问题
    begin
        if @test_3=0 and @str is not NULL
            insert into #imp(cy_imp) values(@str)      --加入不可能表
        set @str = (select cy_res from #array where step=@step-1)
        if (select count(*) from #imp where cy_imp=@str)=0 and @str is not NULL
            insert into #imp(cy_imp) values(@str)
        set @step = @step-2      --返回到两步之前
        set @str = (select cy_res from #array where step=@step)
        set @ch = substring(reverse(@str),patindex('%[吖-咗]%',reverse(@str)),1)
        delete from #array where step>@step   --删除中间的记录
    end
    else
            insert into #array(step, cy_res,num) values(@step, @str, @test_1)
```

第二步，建立以@ch 结尾的词表的游标，然后逐条处理。每次获取结果赋值给@str，然后利用@ch 再赋值给@str 的结尾字，查询在 idioms 表中是否有以@ch

开头的成语，如果有，则退出游标遍历，将该词添加到结果表中；如果不存在，则游标指向下一个词，直到结果不为 0 为止

```
declare cur  scroll cursor for                          --定义游标，寻找成语表中开头字等于当前结尾字的成语
        select cy from idioms where substring(cy,patindex('%[吖-咗]%',cy),1)=@ch
open cur
fetch first from cur into @str  --从第一条结果开始取，作为当前成语
while @@FETCH_STATUS=0
begin
    set @ch = substring(reverse(@str),patindex('%[吖-咗]%',reverse(@str)),1)--更新当前结尾字
    select @test_1 = count(*) from idioms
        where substring(cy,patindex('%[吖-咗]%',cy),1)=@ch
    select @test_2 = count(*) from #array
        where cy_res = @str
    select @test_3 = count(*) from #imp
        where cy_imp = @str
    if @test_1>0 and  @test_2=0 and @test_3=0 and @str is not NULL--如果既未在临时表#array中出现，也满足在结尾字能在成语库中找到匹配的开头字，就退出游标
        break
    else                                --反之，利用游标找到下一个成语作为当前成语，继续寻找
        fetch next from cur into @str
    end
    close cur                           --关闭游标
    deallocate global cur               --释放游标
    set @step = @step + 1               --前进一层
```

最后，删除两个临时表，释放内存

```
select step, cy_res from #array                 --显示结果
order by step                           --按照层数排序
drop table #array                       --删除临时表
select * from #imp                      --查看不可能表
drop table #imp                         --删除不可能表
```

## （二）程序二

拼音库存在同样的问题，即空格很难处理。我们利用和之前类似的思想，找到“第一次出现非 a-z”之前的字符串，就是要找的第一个拼音。即

```
substring(py,1,patindex('%[^a-z]%',py))=@ch_py
```

最后一个字符同理，将内容改成 reverse 就好了。在数据库中有些成语并没有对应的拼音，对于这类词语，我们只能够从 pinyin 库中找到汉字对应的拼音来表示。代码如下所示

```
set @ch = substring(reverse(@str),patindex('%[吖-咗]%',reverse(@str)),1)--更新当前结尾字
set @ch_py = (select top 1 PYM from pinyin where HZ=@ch)
```

其他部分的代码同程序一类似，这是将查询条件改为拼音查询即可。详细程序见程序文件夹中“2_同音.sql”

## （三）程序三

这部分程序主要是寻找从不忘初心到牢记使命的路径。首先将@n 设置为 1000，终止条件设置为@step<1000 或者@ch_py 为’lao’，即与牢同音。详细见程序文件夹中“3_不忘初心_牢记使命.sql”

## （四）程序四

该程序作用是寻找不忘初心到牢记使命的最短路径。最标准的思路应该是采用最短路径的算法进行计算，但由于空间复杂度较高，且 SQL 语言不易编写该算法，我们采用枚举的方法来解决该问题。首先假设中间只有一个词，即开头字拼音为“xin”，结尾字拼音为“lao”的成语，

```
select * from idioms
where    trim(reverse(substring(reverse(py),1,patindex('%[^a-z]%',reverse(py)))))='lao'
and substring(py,1,patindex('%[^a-z]%',py))='xin'
```

发现查询结果为空。这说明不存在这样的词。接下来枚举中间有两个词的答案，即第一个词 a 满足 a 的开头字拼音为“xin”，第二个词 b 满足 b 的结尾字的拼音为“lao”，并且 a 的结尾字的拼音与 b 的开头字的拼音相同。代码如下

```
select a.cy, b.cy
from idioms a, idioms b
where substring(b.py,1,patindex('%[^a-z]%',b.py))=trim(reverse(substring(reverse(a.py),1,patindex('%[^a-z]%',reverse(a.py)))))
and    b.py!=' 无 '    and    substring(a.py,1,patindex('%[^a-z]%',a.py))='xin'    and trim(reverse(substring(reverse(b.py),1,patindex('%[^a-z]%',reverse(b.py)))))='lao'
```

发现查询结果不为空，其结果如本文结果展示部分所示。故最短路径为中间有两个词的情况。