

Re: Please, please, please Help !!!

## Re: Please, please, please Help !!!

---

*Source:*

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.xml/2006-09/msg00001.html>

---

- *From:* Natasha <[Natasha@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Natasha@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 31 Aug 2006 15:07:01 -0700
- 

Hi Peter,

I greatly appreciate your help.

I just tried to follow your logic in the example below, but got completely lost. Could you please take a look when you have a minute.

---

I have a set of 100,000 ascii strings, up to 255 chars each.

Each string has 1 or more words (tokens), space-separated.

A query is input from stdin (1 or more ascii words (tokens), space-separated)

I needed to write pseudocode that determines if the query "soft matches" to any string from (1). By "soft match", I mean that a contiguous subset of tokens from the query must match the entirety of the tokens from a single entry in (1), in the same token order.

For example,

- a. if I have strings in (1): mary poppins, brad pitt, yygr
  - b. and the user types in pictures of brad pitt ---the output should be "true" (because it soft-matches to "brad pitt") or
  - c. if the user types in: brad ---false
  - d. or if the user types in: brad pitt ---true (exactly matches "brad pitt")
  - e. or if the user types in: pitt brad pictures ---false (right tokens as in "brad pitt", but wrong order)
  - f. or if the user types in: brad pitts ---false (char match to "brad pitt", but not a token match)
  - g. or if the user types in: brad yygr ---true (contains "yygr")
- 

Thank you very very much.

Natasha.

"Peter Flynn" wrote:

Natasha wrote:

Hi Peter,

Thak you very much.

Re: Please, please, please Help !!!

Re: Please, please, please Help !!!

But will that return a "yygr" as a result where the "ry" combination is in reversed order?

Yes, because the command it generates looks for each letter separately.

```
$ echo yr|sed -e "s+\(.\)+\1 +g"|tr '\040' '\012'|sort|uniq|grep '[A-Za-z]|awk  
'BEGIN {ORS="";print "cat strings"} {print "|grep -i " $0}'|sh  
mary  
yygr  
$
```

///Peter

Thanks again.  
Natasha.

"Peter Flynn" wrote:

Natasha wrote:

Could anyone please help me:

I have 10000 ascii strings (such as perhaps loaded from a file)

A string is input from stdin.

How to write pseudocode that returns (to stdout) a subset of strings in the file that contain the same distinct characters (regardless of order) as input in.

How to optimize for time.

Assume that this function will need to be invoked repeatedly

For example, if I have strings in: mary, brad, pitt, yygr and the user types

in: ry --> the output should be "mary" and

"yygr" or if the user types in: dd

--> brad

Assume file "strings" (or it could be a pipe or stream) which contains your data:

```
mary  
brad
```

Re: Please, please, please Help !!!

Re: Please, please, please Help !!!

pitt  
yygr

Then accept some input from the user or process, split it into separate characters, get rid of duplicates and non-alphas, and construct a series of Regular Expression to implement the search:

```
$ INPUT=ry  
$ echo $INPUT|sed -e "s+\(.\)+\1 +g"|tr "\040'  
\012'|sort|uniq|grep  
'[A-Za-z]'"|awk 'BEGIN {ORS=""};print "cat strings"} {print  
"|grep -i "  
$0}'|sh
```

(that's all one continuous line). I just tested this on a 100,000-string file and execution was sub-second on an old PIII laptop.

///  
Peter