

Regular Expression in SQL Server

Madhivanan

www.facebook.com/groups/cssug/



Definition

Regular expressions provide better way of matching strings of text, such as particular characters, words, or patterns of characters.

Also referred as

- Regex
- Regexp

How to use Regular expression in SQL Server?

- **A square bracket expression.**
- [abc] matches "a", "b", or "c". [a-z] specifies a range which matches any lowercase letter from "a" to "z". These forms can be mixed: [abcdm-p] matches "a", "b", "c", "d", "m", "n", "o" or "p".
- [ABC] matches "A", "B", or "C". [A-Z] specifies a range which matches any uppercase letter from "A" to "Z". These forms can be mixed: [ABCDM-P] matches "A", "B", "C", "D", "M", "N", "O", or "P"
- [0-9] matches any digit from 0 to 9
- [67] matches digit either 6 or 7
- [^g] matches any character other than g
- [^7-9] matches any character other than digits 7,8 and 9

Pseudo code

- **To check if data has first four characters as alphabets**

where col like '[a-z][a-z][a-z][a-z]%'

- **To check if data has first four characters as alphabets followed by two digits followed by alphabet S**

where col like '[a-z][a-z][a-z][a-z][0-9][0-9][S]%'

- **To check if data ends with alphabet**

where col like '%[a-z]'

Examples with sample data

- **To check if data has atleast a digit**

where col like '%[0-9]%'

- **Only numerics**

where data not like '%[^0-9]%'

Example code

```
declare @sample_table table (data varchar(100) primary key)
insert into @sample_table (data)
select 'This is my new car' union all
select 'Be careful' union all
select 'This red car is very nice' union all
select 'carpets are there' union all
select 'Nothing' union all
select 'Ok. Thank you' union all
select 'My address is No 34, New Main Road, Chennai - 600045' union
all
select 'Please call me at 9444572716' union all
select '9444572716'
```

Query

Select * from @sample_table

Result

This is my new car

Be careful

This red car is very nice

carpets are there

Nothing

Ok. Thank you

My address is No 34, New Main Road, Chennai - 600045

Please call me at 9444572716

9444572716

--Give data that has the word car

```
select * from @sample_table  
where data like '%car%'
```

Result

This is my new car

Be careful

This red car is very nice

carpets are there

--Give data that has car immediately followed by atleast 4 characters

```
select * from @sample_table  
where data like '%car[a-z][a-z][a-z][a-z]%'
```

--or

```
select * from @sample_table  
where data like '%car[a-za-za-za-z]%'
```

Result

Be careful
carpets are there

--Give data that has second letter as either a or l

```
select * from @sample_table  
where data like '_[al]%'
```

Result

carpets are there

Please call me at 9444572716

**--Give data that has second letter as either a or l
and fifth letter is e**

```
select * from @sample_table  
where data like '_[al]__[e]%'
```

Result

carpets are there

--Give data that has eleventh letter as a,n or p

```
select * from @sample_table  
where data like '_____ [anp]%'
```

Result

This red car is very nice

--Get data that has at least a digit

```
select * from @sample_table  
where data like '%[0-9]%'
```

Result

My address is No 34, New Main Road, Chennai - 600045
Please call me at 9444572716
9444572716

--Give data that has number not preceded by hyphen and space

```
select * from @sample_table  
where data like '%[0-9]%' and data not like '%- [0-9]%'
```

Result

9444572716

Please call me at 9444572716

--Get data that has not only digits but any characters too

```
select * from @sample_table  
where data like '%[^0-9]%'
```

Result

This is my new car

Be careful

This red car is very nice

carpets are there

Nothing

Ok. Thank you

My address is No 34, New Main Road, Chennai - 600045

Please call me at 9444572716

--Get data that has only digits

```
select * from @sample_table  
where data not like '%[^0-9]%'
```

Result

9444572716

--Get data that ends with letter e

```
select data from @sample_table  
where data like '%[e]'
```

Result

carpets are there

This red car is very nice

More examples

Example1

**Extract data that has number in the format
Ddd-ddd-dddd (d denotes a number from 0 to 9)**

```
declare @sample_data table(data varchar(100))
insert into @sample_data(data)
select 'I am on the way. Note my number is 87-883-0114 and right now.....'
      as data union all
select 'Well done. Contact this number 345-245-9871 and let me know.....'
```

Select * from @sample_data

Result

I am on the way. Note my number is 87-883-0114 and right now.....
Well done. Contact this number 345-245-9871 and let me know.....

Query

```
select
    data
from
    @sample_data
where
    patindex('%[0-9][0-9][0-9][-][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]%',data)>0
```

Result

Well done. Contact this number 345-245-9871 and let me know.....

- **Extracting the contact number**

```
select
    substring(data,patindex('%[0-9][0-9][0-9][-][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]%',data),12) as ph_no
from
    @sample_data
where
    patindex('%[0-9][0-9][0-9][-][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]%',data)>0
```

Result

345-245-9871

Example 2

Extract amount prefixed by \$

```
declare @sample_data table(data varchar(100))
insert into @sample_data(data)
select '9 Lemons cost 67 $99.99 on sale' as fruit union all
select '$5.99 Apples 877 are on sale' union all
select 'Where are the $65.99 lemons 7856' union all
select 'Oranges costs $99.5' union all
select ' and this costs 98.24'
```

Select * from @sample_data

Result

9 Lemons cost 67 \$99.99 on sale

\$5.99 Apples 877 are on sale

Where are the \$65.99 lemons 7856

Oranges costs \$99.5

and this costs 98.24

Get data that has \$ as part of it

```
select
    data,
    substring(data,charindex('$',data),len(data)) as amount
from
    @sample_data
where
    data like '%$%'
```

Result

Data	amount
9 Lemons cost 67 \$99.99 on sale	\$99.99 on sale
\$5.99 Apples 877 are on sale	\$5.99 Apples 877 are on sale
Where are the \$65.99 lemons 7856	\$65.99 lemons 7856
Oranges costs \$99.5	\$99.5

Get the amount

```
select data,substring(amount,1,patindex('%[0-9][ ]%',amount+' ')) as amount from  
(  
    select  
        data,  
        substring(data,charindex('$',data),len(data)) as amount  
    from  
        @sample_data  
    where  
        data like '%$%'  
) as t
```

Result

9 Lemons cost 67 \$99.99 on sale	\$99.99
\$5.99 Apples 877 are on sale	\$5.99
Where are the \$65.99 lemons 7856	\$65.99
Oranges costs \$99.5	\$99.5

Benefits of Regular expression

- Better way of pattern matching
- Less coding
- Index usage which improves performance

Less Coding

--Give data that starts with four alphabets

Without using Regular expression

```
select * from @sample_table
```

```
where substring(data,1,1) in ('a','b','c',.....'z') And substring(data,2,1) in ('a','b','c',.....'z')
```

```
And substring(data,3,1) in ('a','b','c',.....'z') And substring(data,4,1) in ('a','b','c',.....'z')
```

Result

This is my new car

This red car is very nice

carpets are there

Nothing

Please call me at 9444572716

Using Regular Expression

```
select * from @sample_table  
where data like '[a-z] [a-z] [a-z] [a-z]%'
```

Result

```
This is my new car  
This red car is very nice  
carpets are there  
Nothing  
Please call me at 9444572716
```

Index usage

Get data where first character is b, c or t

Usual method

```
select * from @sample_table  
where substring(data,1,1) in ('b','c', 't')
```

Result

This is my new car

Be careful

This red car is very nice

carpets are there

Table Scan

The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a query plan for a query. The query is:

```
Query 1: Query cost (relative to the batch): 100%
select * from @sample_table where substring(data,1,1) in ('b','c','t')
```

The query plan shows a single operator: **Clustered Index Scan**. The cost is 100%.

Clustered Index Scan
Scanning a clustered index, entirely or only a range.

Property	Value
Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001581
Estimated Operator Cost	0.0032831 (100%)
Estimated Subtree Cost	0.0032831
Estimated Number of Rows	1
Estimated Row Size	61 B
Ordered	False
Node ID	1

Predicate
substring([data],(1),(1))='t' OR substring([data],(1),(1))='c' OR substring([data],(1),(1))='b'

Object
[@sample_table].[PK_#10818984_1175A0BD]

Output List
data

Regular expression

```
select * from @sample_data  
where data like '[bct]%'
```

Result

This is my new car
Be careful
This red car is very nice
carpets are there

Table Seek

The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a query plan for a query named 'seek.sqlplan'. The query is: `select * from @sample_table where data like '[bct]%'`. The query cost is 100%. The query plan shows a 'Clustered Index Seek' operation with a cost of 100. A tooltip is displayed over the operation, providing detailed information about the seek operation.

Query 1: Query cost (relative to the batch): 100%

```
select * from @sample_table where data like '[bct]%'
```

Clustered Index Seek
Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001581
Estimated Operator Cost	0.0032831 (100%)
Estimated Subtree Cost	0.0032831
Estimated Number of Rows	1
Estimated Row Size	61 B
Ordered	True
Node ID	0

Predicate
[data] like [bct]%'

Object
[@sample_table].[PK__#10818984__1175ADBD]

Output List
data

Seek Predicates
Start Range: data >= Scalar Operator('Ä...
Ä%Ä%Ä%Ä%Ä%Ä%'), End Range: data < Scalar Operator ('U')

Get data where length is >=10

```
select data from @sample_table where len(data)>=10
```

Result

9444572716

Be careful

carpets are there

My address is No 34, New Main Road, Chennai - 600045

Ok. Thank you

Please call me at 9444572716

This is my new car



This red car is very nice

- Execution plan (Index scan)

Messages Execution plan

Query 2: Query cost (relative to the batch): 20%

```
select data from @sample_table where len(data) >= 10
```

 ← 

SELECT
Cost: 0 %

Clustered Index Scan
[@sample_table].[PK__#A1CDEF4__D9DE...]
Cost: 100 %

```
select data from @sample_table
where data like '[A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.]%'
```

Result

9444572716

Be careful

carpets are there

My address is No 34, New Main Road, Chennai - 600045

Ok. Thank you

Please call me at 9444572716

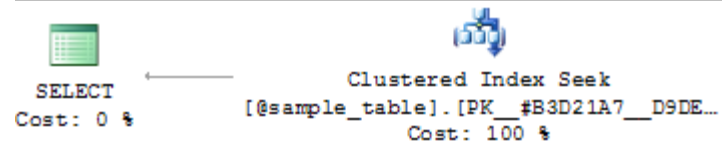
This is my new car

This red car is very nice

- Execution plan (Index seek)

Query 2: Query cost (relative to the batch): 25%

```
select data from @sample_table where data like '[A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.][A-Z0-9 ,.]..'
```



```
select data from @sample_table  
where data like '_____ %'
```

Result

9444572716

Be careful

carpets are there

My address is No 34, New Main Road, Chennai - 600045

Ok. Thank you

Please call me at 9444572716

This is my new car

This red car is very nice

```
create table #test(pan_no char(10) primary key)
insert into #test (pan_no)
select 'ALUJM8765H' union all
select 'GHOKL7653K' union all
select 'IMNK68765H' union all
select 'LOMRF0897U' union all
select 'LPIAC778J' union all
select 'MZXUI1296E'
```


Find out pan no with length =10


```
select pan_no from #test
where len(pan_no)=10
```

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
select pan_no from #test where len(pan_no)=10
```

 SELECT
Cost: 0 %

 Clustered Index Scan (Clustered)
[#test].[PK_#test_93D5B9B2E3A7...]
Cost: 100 %

```
select pan_no from #test
where pan_no like '[A-Z0-9][A-Z0-9][A-Z0-9][A-
Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-
9][A-Z0-9]'
```

- Execution plan (Index seek)

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
select pan_no from #test where pan_no like '[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9] ...
```

SELECT
Cost: 0 %

Clustered Index Seek (Clustered)
[#test].[PK__#test____93DSB9B2E3A7...]
Cost: 100 %

Thank You

Blog : <http://beyondrelational.com/blogs/madhivanan>

Twitter : www.twitter.com/madhivanan2001