

Регулярные выражения

Регулярные выражения это мощный инструмент работы со строками, он позволяет находить подстроки, сравнивать и считать совпадения по маске, которая пишется специальным синтаксисом.

Регулярные выражения встречаются не только в SQL, но и во многих языках программирования.

Мы разберем с вами в первую очередь синтаксис самих регулярных выражений, как правильно формировать маску, а потом рассмотрим функции обработки строк по средствам регулярных выражения.

Группы символов

В регулярных выражениях есть специальные конструкции, которые обозначают те или иные группы символов.

`\w` - буквы и цифры

`\W` - Не буквы и цифры

`\d` - цифры

`\D` - Не цифры

`[abcABCaбвАБВ012]` - пользовательский набор символов

`.` - любой символ

`^` - начало строки

`$` - конец строки

Для демонстрации примеров мы будем использовать функцию `regexp_like`, которая возвращает истину, если маска подходит и ложь, если нет.

```
-- запрос вернет записи с именами, начинающиеся с S или D
-- обратите внимание что в regexp регистр букв важен

select * from hr.employees
where regexp_like(FIRST_NAME, '^[SD]');

-- данный запрос вернет телефон в представленном формате
```

```
-- запрос будет оптимальнее при использовании квантификаторов

select * from hr.employees
where regexp_like(PHONE_NUMBER, '\d\d\d.\d\d\d.\d\d\d\d');

/*
в запросе выше есть неточность, символ точки в данном случае является
частью маски и обозначает один любой символ, для того, чтобы указать именно
точку нам необходимо ее "экранировать" указав до нее обратный слеш
*/

select
*
from hr.employees
where regexp_like(PHONE_NUMBER, '\d\d\d\.\d\d\d\.\d\d\d\d');
```

При поиске символов и цифр (`\w`) есть свои особенности, давайте их разберем.

```
select
*
from (
  select 'привет' as n from dual
  union all
  select 'hello' as n from dual
  union all
  select '345' as n from dual
  union all
  select 'привет hello' as n from dual
  union all
  select 'привет hello' as n from dual
  union all
  select '1234hello' as n from dual)
where regexp_like(n, '\w');

-- \w ищет кириллицу, латиницу и цифры.

select
*
from (
  select 'привет' as n from dual
  union all
  select 'hello' as n from dual
  union all
  select '345' as n from dual
  union all
  select 'привет hello' as n from dual
  union all
  select 'привет hello' as n from dual
  union all
  select '1234hello' as n from dual)
where regexp_like(n, '[a-zA-Я]');
```

```
-- для поиска кириллицы необходимо указывать диапазон (в него не входит буква ё)
-- для поиска слов с буквой ё необходимо ее указывать отдельно [а-яА-ЯёЁ]
```

Задание

Таблица **hr.employees**

1. Найдите пользователей, у которых в имени есть одна из букв (a,f,r,t)
2. Найдите пользователей, у которых имя начинается с одной из букв (a,f,r,t)

```
-- 1
select * from hr.employees
where regexp_like(FIRST_NAME, '[afrt]');

-- 2

select * from hr.employees
where regexp_like(FIRST_NAME, '^[AFRT]');
```

Квантификаторы

Это оператор регулярного выражения, который позволяет указать кол-во символов из группы.

+ - один или больше

* - ноль или много

? - один или ноль

{n} - ровно n

{n,} - n или больше

{n,m} - от n до m

{,m} - от 0 до m

Рассмотрим пример использования квантификатора на заданиях.

Задание

1. Таблица **hr.employees**. Выведите только те записи, в которых номер телефона имеет формат XXX.XXX.XXXX
2. Написать запрос, который выводит только те записи, где вторая часть названия job_id состоит из 2 или 3 символов
3. Создать запрос, который выводит те записи из таблицы oe.PRODUCT_information, в которых в поле PRODUCT_DESCRIPTION указаны размеры товара.
4. Таблица **hr.departments**. Выведите только те записи, у которых название департамента состоит не более, чем из 2 слов
5. Создайте запрос, который позволяет найти строки с корректной электронной почтой.

```
-- 1
select * from hr.employees
where regexp_like(PHONE_NUMBER, '^d{3}\.d{3}\.d{4}$');

-- квантификатор после \d указывает, сколько цифр должно быть подряд
-- в данной строке у нас идет повторение 3 цифр и точки два раза
-- чтобы сделать регулярное выражение еще более коротким мы можем
-- объединить эту сигнатуру в группу (используя скобки) и указать квантификатор
-- у группы

select * from hr.employees
where regexp_like(PHONE_NUMBER, '^(\d{3}\.){2}\d{4}$');

-- 2

select job_id from hr.employees
where regexp_like(job_id, '^w{2}_w{2,3}$');
-- в данном задании квантификатор указывает диапазон от 2 до 3 символов

-- 3
select PRODUCT_DESCRIPTION from oe.PRODUCT_information
where regexp_like(PRODUCT_DESCRIPTION, '\: w+\.?w* x w+\.?w* x w+\.?w*');

-- с использованием групп этот запрос будет иметь следующий вид

select
    PRODUCT_DESCRIPTION
from oe.PRODUCT_information
where regexp_like(PRODUCT_DESCRIPTION, '\: (w+\.?w* x){2} w+\.?w*');

-- 4
select DEPARTMENT_NAME from hr.departments
where regexp_like(DEPARTMENT_NAME, '^(\w+ ?){1,2}$')
```

```
-- в данном случае сигнатура " ?" обозначает, что в конце группы может быть пробел
-- он будет отсутствовать только у последнего значения так как в конце последнего
-- слова пробела нет




--5

select
  *
from (
  select 'hayk.inanc@gmail.com' as n from dual
  union all
  select 'inanc@mail.ru' as n from dual
  union all
  select 'inanc hayk@rambler.ru' as n from dual
  union all
  select 'inanc_hayk@rambler.RU' as n from dual
  union all
  select '89096450730@yahoo.com' as n from dual
) t1
where regexp_like(n, '^[a-zA-Z\.\_0-9\-\]+\@\w+\.[a-zA-Z]{2,4}$')

-- стоит заметить, что это не универсальная регулярка по почтам
-- одна покрывает большинство кейсов
```

Другие regexp функции

В oracle есть целый ряд других функций работы с регулярными выражениями. Давайте их разберем.

 Название	 пример использования	 описание
<u>REGEXP_COUNT</u>	REGEXP_COUNT('1 2 3 abc','\d')	Кол-во совпадений
<u>REGEXP_INSTR</u>	REGEXP_INSTR('Y2K problem','\d+')	Позиция совпадения
<u>REGEXP_LIKE</u>	REGEXP_LIKE('Year of 2017','\d+')	Проверка соответствия
<u>REGEXP_REPLACE</u>	REGEXP_REPLACE('Year of 2017','\d+', 'Dragon')	Замена на подстроку
<u>REGEXP_SUBSTR</u>	REGEXP_SUBSTR('Number 10','\d+')	Нахождение подстроки

REGEXP_COUNT

Данная функция позволяет определить кол-во вхождений подстроки в строку.

Рассмотрим пример

```
-- определить кол-во слов в строке
select REGEXP_COUNT('привет мой дорогой друг', '\w+') from dual;
```

REGEXP_INSTR

Данная функция позволяет определить положение подстроки в строке

```
select
  REGEXP_INSTR('
    это длинный текст, внутри которого есть число, а вот и оно 123, вот!
  ', '\d+')
from dual;
```

REGEXP_LIKE

Я полагаю, эту функцию мы уже разобрали достаточно хорошо =).

REGEXP_REPLACE

Данная функция позволяет заменить подстроку на другую подстроку, очень часто используется при очистке строк.

```
select
  regexp_replace(n, '[_ \-]') as phone
from (
  select '+7-909-645-07-30' as n from dual
  union all
  select '+7_909_645_0730' as n from dual
  union all
  select '+7 90964507 30' as n from dual
  union all
  select '+7 909 6450730' as n from dual
  union all
  select '+7 909 645 07 30' as n from dual
) t1
```

REGEXP_SUBSTR

Данная функция позволяет нам получить подстроку из строки, которая совпадает с маской.

Обратите внимание, что REGEXP_SUBSTR возвращает только первое вхождение.

```
-- найти слово из 6 букв

select
  regexp_substr(n, '(^|\w)[a-zA-Z]{6}($|\w)')
from (
  select
    'Lorem ipsum, dolor sitddd amet consectetur adipisicing elit.' as n
  from dual
) t1;

-- обратите внимание, что сигнатура (^|\w) обозначает начало слова, а ($|\w) конец.
```