

При помощи функции ROW_NUMBER сгенерируем порядковый номер строки запроса.

```
garage=# select row_number() over (order by name) num, name from Owner;
 num |      name
-----+-----
   1 | Aleksey Alekseev
   2 | Ivan Ivanov
   3 | Petr Petrov
(3 rows)
```

```
garage=# select * from ( select row_number() over (order by name) num, name from Owner) Owner where num < 3;
 num |      name
-----+-----
   1 | Aleksey Alekseev
   2 | Ivan Ivanov
(2 rows)
```

Функция ABS(n) возвращает абсолютное значение числа n

```
garage=# select abs(-500) x1, abs(500) x2;
 x1 | x2
-----+-----
 500 | 500
(1 row)
```

Функция FLOOR(n) возвращает наименьшее целое, меньшее или равное переданному в качестве параметра числу n.

```
garage=# select floor(100.1) x1, floor(-100.1) x2, floor(100.9) x3;
 x1 | x2 | x3
-----+-----+-----
 100 | -101 | 100
(1 row)
```

Функция TRUNC(n, m) возвращает число n, усеченное до m знаков после десятичной точки.

```
garage=# select trunc(100.123, 1) x1, trunc(100.123) x2, trunc(100.123, 4) x3;
 x1 | x2 | x3
-----+-----+-----
 100.1 | 100 | 100.1230
(1 row)
```

Функция ROUND(n[,m]) возвращает число n, округленное до m знаков после десятичной точки по правилам математического округления

```
garage=# select round(100.123, 1) x1, round(100.123) x2, trunc(100.125, 2) x3;
 x1 | x2 | x3
-----+-----+-----
 100.1 | 100 | 100.12
(1 row)
```

Функция SIGN(n) определяет знак числа.

```
garage=# select sign(100) x1, sign(-100) x2, sign(0) x3;
 x1 | x2 | x3
----+---+---
  1 | -1 |  0
(1 row)
```

Функция MOD(n, m) возвращает остаток от деления n на m

```
garage=# select mod(100, 24) x1, mod(100, 25) x2, mod(100, 23) x3;
 x1 | x2 | x3
----+---+---
  4 |  0 |  8
(1 row)
```

Функция POWER(n, m) возводит число n в степень m

```
garage=# select power(10, 4) x1, power(2, 3) x2, power(100, 0.5) x3;
 x1 | x2 | x3
----+---+---
10000 | 8 | 10.0000000000000000
(1 row)
```

Функция SQRT(n) возвращает квадратный корень от числа n.

```
garage=# select sqrt(100) x1, sqrt(9) x2;
 x1 | x2
----+---
 10 |  3
(1 row)
```

Функция EXP(n) возводит e в степень n, а функция LN(n) вычисляет натуральный логарифм от n

```
garage=# select exp(1) x1, ln(1) x2, ln(exp(3)) x3;
 x1 | x2 | x3
----+---+---
2.718281828459045 | 0 | 3
(1 row)
```

Функция LOG(n, m) производит вычисление логарифма m по основанию n.

```
garage=# select log(10, 100) x1, log(25, 5) x2;
 x1 | x2
----+---
2.0000000000000000 | 0.5000000000000000
(1 row)
```

Функции Sin(n), Cos(n), Tan(n), Cot(n) производят вычисление тригонометрических функций.

```
garage=# select sin(1) x1, cos(0) x2, tan(1) x3, cot(0) x4;
          x1          | x2 |          x3          | x4
-----+-----+-----+-----
0.8414709848078965 | 1 | 1.5574077246549023 | Infinity
(1 row)
```

Функция CONCAT(str1, str2) выполняет склеивание строк str1 и str2

```
garage=# select concat('str', 'ing') x1;
          x1
-----
string
(1 row)
```

Функция LOWER(str) преобразует все символы строки str в строчные

```
garage=# select lower('TeSt StRING') x1;
          x1
-----
test string
(1 row)
```

Функция UPPER(str) преобразует все символы строки str в прописные.

```
garage=# select upper('TeSt UpPeR') x1;
          x1
-----
TEST UPPER
(1 row)
```

Функция INITCAP(str) возвращает строку str, в которой первые буквы всех слов преобразованы в прописные

```
garage=# select initcap('iVAN iVanoV') x1;
          x1
-----
Ivan Ivanov
(1 row)
```

Функция LTRIM(str, [,set]) удаляет все символы с начала строки до первого символа, которого нет в наборе символов set.

```
garage=# select ltrim('abcdef', 'abdef') x1;
          x1
-----
cdef
(1 row)
```

Функция RTRIM(str, [,set]) аналогична, но удаляет символы, начиная от конца строки

```
garage=# select rtrim('abcdef', 'abdef') x1;
x1
-----
abc
(1 row)
```

Функция REPLACE(str, search_str, replace_str) осуществляет поиск образца search_str в строке str и каждое найденное вхождение заменяет на replace_str.

```
garage=# select replace('Test upper function', 'upper', 'replace') x1;
x1
-----
Test replace function
(1 row)
```

Функция TRANSLATE(str, from_mask, to_mask) анализирует строку str и заменяет в ней все символы, встречающиеся в строке from_mask, на соответствующие символы из to_mask

```
garage=# select translate('Еуые translaey funceion', 'еуЕ', 'tesT') x1;
x1
-----
Test translate function
(1 row)
```

Функция LENGTH(str) возвращает длину строки str в символах.

```
garage=# select length('Test length function') x1;
x1
-----
20
(1 row)
```

Функция ASCII(str) возвращает ASCII-код первого символа строки str в случае применения кодировок ASCII и UTF-8.

```
garage=# select ASCII('Test') x1, ASCII(' ') x2;
x1 | x2
----+----
84 | 32
(1 row)
```

Функция CHR(n) возвращает символ по его коду

```
garage=# select chr(84) x1, chr(101) x2, chr(115) x3, chr(116) x4;
 x1 | x2 | x3 | x4
-----+-----+-----+-----
  T | e  | s  | t
(1 row)
```

Функция NOW() возвращает текущую дату и время по часам сервера

```
garage=# select now();
      now
-----
2023-05-17 14:49:08.784789-04
(1 row)
```

Функция JUSTIFY_INTERVAL(interval) преобразует интервал, указанный в виде строки в соответствующее значение

```
garage=# select now(), now() + justify_interval('5 days 0 hours 40 minute');
      now      |      ?column?
-----+-----
2023-05-17 14:50:22.833112-04 | 2023-05-22 15:30:22.833112-04
(1 row)
```

Функция DATE_TRUNC(timestamp) используется для обрезки даты или интервала (DATE_TRUNC(interval)) до определенной точности

```
garage=# select date_trunc('minute', now()), date_trunc('year', now());
      date_trunc      |      date_trunc
-----+-----
2023-05-17 14:51:00-04 | 2023-01-01 00:00:00-05
(1 row)
```

Функция AGE([end_date,]start_date) возвращает разницу между датами, обозначенными как end_date и start_date.

```
garage=# select current_date d1, age(make_timestamp(2020, 4, 15, 4, 24, 00)) d2, age(make_date(2023, 12, 12), make_timestamp(2020, 4, 15, 4, 24, 00)) d3;
 d1 |      d2      |      d3
-----+-----+-----
2023-05-17 | 3 years 1 mon 1 day 19:36:00 | 3 years 7 mons 26 days 19:36:00
(1 row)
```

Функция EXTRACT(field FROM timestamp) извлекает элемент даты field из значения типа timestamp

```
garage=# select now(), extract (day from now()), extract(minute from now());
      now      |      extract      |      extract
-----+-----+-----
2023-05-17 14:57:46.237378-04 |      17      |      57
(1 row)
```

Функция TO_DATE(str, mask) преобразует строку str в дату.

```
garage=# select to_date ('24 jun 2020', 'dd mon yyy');
 to_date
-----
2020-06-24
(1 row)
```

Функция TO_CHAR(date, mask) преобразует дату date в символьную строку в соответствии с заданной маской

```
garage=# select to_char(now(), 'dd.mm.yy');
 to_char
-----
17.05.23
(1 row)
```