Функция ROW_NUMBER генерирует порядковый номер строки запроса.

```
photo=# select row_number() over (order by name) num, name from client;
 num |                name
-----+-------------------------------------
   1 | Каркаров Игорь Петрович
   2 | Костяшкин Кирилл Филиппович
   3 | Молякова Светлана Михайловна
   4 | Проскурина Анна Владимировна
(4 строки)
```

Также ROW_NUMBER может применяться для ограничения количества обрабатываемых строк.

```
photo=# select * from (select row_number() over (order by name) num, name from client) client where num<3;
 num |               name
-----+-----------------------------------
   1 | Каркаров Игорь Петрович
   2 | Костяшкин Кирилл Филиппович
(2 строки)
```

Функция ABS(n) возвращает абсолютное значение числа n.

```
photo=# select abs(1843) X1, abs(-47) X2, abs(-583) X3;
  x1  | x2 | x3
------+----+-----
 1843 | 47 | 583
(1 строка)
```

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

```
photo=# select ceil(48.2) X1, ceil(-48.2) X2, ceil(47.8) X3, ceil(-47.8) X4;
 x1 | x2  | x3 | x4
----+-----+----+-----
 49 | -48 | 48 | -47
(1 строка)
```

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

```
photo=# select floor(48.2) X1, floor(-48.2) X2, floor(47.8) X3, floor(-47.8) X4;
 x1 | x2  | x3 | x4
----+-----+----+-----
 48 | -49 | 47 | -48
(1 строка)
```

Функция TRUNC(n, m) возвращает число n, усеченное до m знаков после десятичной точки. Параметр m может не указываться – в этом случае n усекается до целого.

```
photo=# select trunc(48.258393, 3) X1, trunc(-48.258393) X2, trunc(47.8657) X3, trunc(-47.8657, 2) X4;
   x1   | x2  | x3 |   x4
--------+-----+----+--------
 48.258 | -48 | 47 | -47.86
(1 строка)
```

Функция ROUND(n, m) возвращает число n, округленное до m знаков после десятичной точки по правилам математического округления. Параметр m может не указываться – в этом случае n округляется до целого.

```
photo=# select round(48.258393) X1, round(-48.258393) X2, round(47.8657) X3, round(-47.8657, 3) X4;
 x1 | x2  | x3 |   x4
----+-----+----+---------
 48 | -48 | 48 | -47.866
(1 строка)
```

Функция SIGN(n) определяет знак числа. Если n положительное, то функция возвращает 1. Если отрицательное – возвращается – 1. Если равно нулю, то возвращается 0.

```
photo=# select sign(1843) X1, sign(0) X2, sign(-583) X3;
 x1 | x2 | x3
----+----+----
  1 |  0 | -1
(1 строка)
```

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

```
photo=# select mod(1843, 7) X1, mod(10, 2) X2, mod(-585, 5) X3;
 x1 | x2 | x3
----+----+----
  2 |  0 |  0
(1 строка)
```

Функция POWER(n, m) возводит число n в степень m. Степень может быть дробной и отрицательной, что существенно расширяет возможности данной функции.

```
photo=# select power(3, 3) X1, power(3, -3) X2, power(4, 0.5) X3, power(100, -0.333) X4;
 x1 |          x2          |         x3         |         x4
----+----------------------+--------------------+--------------------
 27 | 0.037037037037037035 | 2.0000000000000000 | 0.2157744409152666
(1 строка)
```

В некоторых случаях при вызове данной функции может возникнуть исключительная ситуация. В данном случае производится попытка вычисления квадратного корня от отрицательного числа, что приведет к возникновению ошибки «Отрицательное число в дробной степени дает комплексный результат».

```
photo=# select power(-4, 0.5);
ОШИБКА:  отрицательное число в дробной степени даёт комплексный результат
```

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

```
photo=# select sqrt(625);
 sqrt
------
   25
(1 строка)
```

Функция EXP(n) возводит *e* в степень n, а функция LN(n) вычисляет натуральный логарифм от n (при этом значение n должно быть больше нуля).

```
photo=# select exp(3) X1, ln(1) X2, ln(exp(2)) X3;
         x1         | x2 | x3
--------------------+----+----
 20.085536923187668 |  0 |  2
(1 строка)
```

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

```
photo=# select log(4, 64);
        log
--------------------
 3.0000000000000000
(1 строка)
```

PostgreSQL поддерживает вычисление основных тригонометрических функций:
• *SIN(n)* – синус *n* (где *n* – угол в радианах);
• *COS(n)* – косинус *n* (где *n* – угол в радианах);
• *TAN(n)* – тангенс *n* (где *n* – угол в радианах);
• *COT(n)* – котангенс *n* (где *n* – угол в радианах).

```
photo=# select sin(1) X1, cos(1) X2, tan(0) X3, cot(0) X4;
         x1         |         x2         | x3 |    x4
--------------------+--------------------+----+----------
 0.8414709848078965 | 0.5403023058681398 |  0 | Infinity
(1 строка)
```

Функция CONCAT(str1, str2) выполняет конкатенацию строк str1 и str2. Если один из аргументов равен NULL, то он воспринимается как пустая строка. Если оба аргумента равны NULL, то функция возвращает NULL.

```
photo=# select concat('Он ', 'любил') X1, concat('Он ', null) X2, concat(null, 'любил') X3, concat(null, null) X4;
    x1    | x2  |  x3   | x4
----------+-----+-------+----
 Он любил | Он  | любил |
(1 строка)
```

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

```
photo=# select lower('Я УСТАЛА');
  lower
----------
 я устала
(1 строка)
```

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

```
photo=# select upper('я устала');
  upper
----------
 Я УСТАЛА
(1 строка)
```

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

```
photo=# select initcap('у попА былА сОбакА');
      initcap
--------------------
 У Попа Была Собака
(1 строка)
```

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

```
photo=# select replace('У попа была еда', 'еда', 'собака');
      replace
--------------------
 У попа была собака
(1 строка)
```

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.

```
photo=# select translate('У роns быкф сovфса', 'pskфvфс', 'палабак');
     translate
--------------------
 У попа была собака
(1 строка)
```

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

```
photo=# select length('Он ее любил');
 length
--------
     11
(1 строка)
```

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

```
photo=# select ascii('Он ее любил');
 ascii
-------
  1054
(1 строка)
```

Функция CHR(n) возвращает символ по его коду.

```
photo=# select chr(89), chr(96), chr(104);
 chr | chr | chr
-----+-----+-----
 Y   | `   | h
(1 строка)
```

Функция NOW() - это одна из самых часто употребляемых функций, она возвращает текущую дату и время по часам сервера.

```
photo=# select now();
              now
-------------------------------
 2023-05-02 12:39:15.143015+03
(1 строка)
```

Функция JUSTIFY_INTERVAL(interval) преобразует интервал (тип interval), указанный в виде строки в соответствующее значение типа timestamp.

```
photo=# select now(), now() + justify_interval ('14 days 2 hour 10 minute');
              now               |           ?column?
--------------------------------+-------------------------------
 2023-05-02 12:44:42.33326+03 | 2023-05-16 14:54:42.33326+03
(1 строка)
```

Функция AGE(end_date, start_date) возвращает разницу между датами, обозначенными как end_date и start_date. Если параметр end_date опущен, то используется значение глобальной переменной CURRENT_DATE, которая содержит текущую дату (тип date, дата без времени).

```
photo=# select current_date d1, age(make_timestamp(2023, 6, 1, 10, 5, 40)) d2, age(make_date(2023, 6, 1), make_timestamp(2023, 8, 31, 10, 5, 40)) d3;
     d1     |        d2         |            d3
------------+-------------------+--------------------------
 2023-05-02 | -30 days -10:05:40 | -2 mons -30 days -10:05:40
(1 строка)
```

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

```
photo=# select now(), extract (month from now()), extract (minute from now());
              now               | extract | extract
--------------------------------+---------+---------
 2023-05-02 12:58:10.235746+03 |       5 |      58
(1 строка)
```

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

```
photo=# select to_date ('1 jun 2023', 'dd mon yyy');
  to_date
------------
 2023-06-01
(1 строка)
```

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

```
photo=# select to_char (now(), 'dd.mm.yyyy');
  to_char
------------
 02.05.2023
(1 строка)
```