

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**ФГБОУ ВО «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

**БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ**

**ПРАКТИЧЕСКАЯ РАБОТА №5**

**ВЫПОЛНИЛА: Юрьева Марина  
ГРУППА: УБ-02**

**ВОРОНЕЖ  
2023**

При помощи функции ROW\_NUMBER сгенерируем порядковый номер строки запроса.

```
аптека=# SELECT ROW_NUMBER() OVER (ORDER BY fio) Num, fio from Buyers;
num |      fio
-----+-----
  1 | Аистова Елизавета Андреевна
  2 | Безотосный Илья Игоревич
  3 | Моисеев Вадим Олегович
  4 | Некрасова Мария Ивановна
  5 | Самцова Елизавета Михайловна
  6 | Соболева Елизавета Сергеевна
  7 | Старухин Александр Геннадьевич
(7 строк)
```

```
аптека=# SELECT * FROM (SELECT ROW_NUMBER () OVER (ORDER BY fio) NUM, fio FROM Buyers) Buyers WHERE NUM <=5;
num |      fio
-----+-----
  1 | Аистова Елизавета Андреевна
  2 | Безотосный Илья Игоревич
  3 | Моисеев Вадим Олегович
  4 | Некрасова Мария Ивановна
  5 | Самцова Елизавета Михайловна
(5 строк)
```

Функция ABS(n) возвращает абсолютное значение числа n.

```
аптека=# SELECT ABS(-100) X1, ABS(100) X2;
x1 | x2
-----+-----
100 | 100
(1 строка)
```

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

```
аптека=# SELECT CEIL(-200) X1, CEIL(200) X2, CEIL(205.4) X3, CEIL(-200.1) X4;
x1 | x2 | x3 | x4
-----+-----+-----+-----
-200 | 200 | 206 | -200
(1 строка)
```

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

```
аптека=# SELECT FLOOR (103.2) X1, FLOOR (-201.99) X2;
x1 | x2
-----+-----
103 | -202
(1 строка)
```

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

```
аптека=# SELECT TRUNC(100.123456, 2) X1, TRUNC(-100.32256) X2;  
x1 | x2  
-----+-----  
100.12 | -100  
(1 строка)
```

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

```
аптека=# SELECT ROUND (100.123456) X1, ROUND (-100.32256, 4) X2;  
x1 | x2  
-----+-----  
100 | -100.3226  
(1 строка)
```

Функция SIGN(n) определяет знак числа.

```
аптека=# SELECT SIGN (100) X1, SIGN (0) X2, SIGN (-100) X3;  
x1 | x2 | x3  
----+----+----  
1 | 0 | -1  
(1 строка)
```

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

```
аптека=# SELECT MOD (500, 10) X1, MOD (100, 13) X2;  
x1 | x2  
----+----  
0 | 9  
(1 строка)
```

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

```
аптека=# SELECT POWER (2, 8) X1, POWER (10, 0.5) X2, POWER (20, -0.75) X3;  
x1 | x2 | x3  
-----+-----+-----  
256 | 3.1622776601683793 | 0.1057371263440564  
(1 строка)
```

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

```
аптека=# SELECT SQRT(256) X1, SQRT(121) X2;
  x1 | x2
-----+-----
  16 | 11
(1 строка)
```

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

```
аптека=# SELECT EXP(2) X1, LN(8) X2, LN(EXP(6)) X3;
      x1      |      x2      |      x3
-----+-----+-----
7.38905609893065 | 2.0794415416798357 | 6
(1 строка)
```

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

```
аптека=# SELECT LOG(2, 4) X1;
      x1
-----
2.0000000000000000
(1 строка)
```

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

```
аптека=# SELECT SIN(4) X1, COS(6) X2, TAN(2) X3, COT(1) X4;
      x1      |      x2      |      x3      |      x4
-----+-----+-----+-----
-0.7568024953079282 | 0.960170286650366 | -2.185039863261519 | 0.6420926159343306
(1 строка)
```

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

```
артека=# SELECT LOWER ('ПОСТАВЬТЕ ЗАЧЕТ');
        lower
-----
    поставьте зачет
(1 строка)
```

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

```
артека=# SELECT UPPER ('поставьте зачет');
        upper
-----
    ПОСТАВЬТЕ ЗАЧЕТ
(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.

```
артека=# SELECT REPLACE ('Музыка играет из рыбы', 'рыбы', 'колонок');
        replace
-----
    Музыка играет из колонок
(1 строка)
```

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

```
артека=# SELECT LENGTH ('Музыка играет из колонок');
        length
-----
            24
(1 строка)
```

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

```

apteka=# SELECT ASCII ('Музыка');
 ascii
-----
    1052
(1 строка)

```

Функция CHR(n) возвращает символ по его коду.

```

apteka=# SELECT CHR(100), CHR(111), CHR(109);
 chr | chr | chr
-----+-----
  d  | o  | m
(1 строка)

```

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

```

apteka=# SELECT NOW();
          now
-----
2023-05-25 02:37:16.148638+03
(1 строка)

```

Функция JUSTIFY\_INTERVAL(interval) преобразует интервал, указанный в виде строки в соответствующее значение.

```

apteka=# SELECT NOW(), NOW() + JUSTIFY_INTERVAL ('4 DAYS 3 HOUR 6 MINUTE');
          now |
-----+-----
2023-05-25 02:40:55.794475+03 | 2023-05-29 05:46:55.794475+03
(1 строка)

```

Функция DATE\_TRUNC(timestamp) используется для обрезки даты или интервала (DATE\_TRUNC(interval)) до определенной точности.

```

apteka=# SELECT DATE_TRUNC ('DAY', NOW()), DATE_TRUNC('MINUTE', NOW());
 date_trunc | date_trunc
-----+-----
2023-05-25 00:00:00+03 | 2023-05-25 02:43:00+03
(1 строка)

```

Функция TO\_DATE(str, mask) преобразует строку str в дату.

```
apтека=# SELECT TO_DATE ('28 MAY 2002', 'DD Mon YYYY');  
to_date  
-----  
2002-05-28  
(1 строка)
```

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

```
apтека=# SELECT TO_CHAR (NOW(), 'DD.MM.YY');  
to_char  
-----  
25.05.23  
(1 строка)
```