



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ**

БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

Отчет

по практической работе 5

«Дополнительные возможности POSTGRESQL»

Отчет подготовил
студент 3 курса группы УБ-01
Хомутов Константин

Воронеж 2023

5. Дополнительные возможности POSTGRESQL

5.1 Функция ROW_NUMBER

Функция ROW_NUMBER() генерирует порядковый номер строки запроса.

```
itco2=# SELECT ROW_NUMBER() OVER (ORDER BY name) num, name FROM student;
```

num	name
1	Капустина Влада Мироновна
2	Кудрявцев Альберт Миронович
3	Мышкина Дария Кимовна
4	Николаев Терентий Вячеславович
5	Тетерин Климент Денисович

(5 строк)

Так же ROW_NUMBER может применяться для ограничения количества обрабатываемых строк:

```
SELECT * FROM (  
    SELECT ROW_NUMBER() OVER (ORDER BY name) num, name  
FROM student ) student WHERE num <= 3;
```

```
itco2=# SELECT * FROM (  
itco2=# SELECT ROW_NUMBER() OVER (ORDER BY  
num | name  
-----+-----  
1 | Капустина Влада Мироновна  
2 | Кудрявцев Альберт Миронович  
3 | Мышкина Дария Кимовна  
(3 строки)
```

5.2 Функция COALESCE

Функция COALESCE возвращает значение первого аргумента, значение которого не равно NULL. Функция возвращает NULL в том случае, если все аргументы имеют значение NULL.

SELECT grade, COALESCE(grade, 0) Coalesce_Grade FROM grade;

```
itco2=# SELECT grade, COALESCE(grade, 0) C
grade | coalesce_grade
-----+-----
      80 |              80
      90 |              90
      75 |              75
      60 |              60
      85 |              85
      65 |              65
      93 |              93
     100 |             100
      90 |              90
      90 |              90
         |              0
         |              0
(12 строк)
```

5.3 Числовые функции

Функция ABS

Функция ABS(n) возвращает абсолютное значение числа n.

SELECT ABS(1) A, ABS(-1) B, ABS(-1.2) C;

```
itco2=# SELECT ABS(1) A, ABS(-1) B, ABS(-1.2) C;
 a | b | c
---+---+---
  1 | 1 | 1.2
(1 строка)
```

Функция CEIL

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

SELECT CEIL(10) A, CEIL(10.5) B, CEIL(-10.5) C, CEIL(-10) D;

```
itco2=# SELECT CEIL(10) A, CEIL(10.5) B, CEIL (-10.5) C, CEIL(-10) D;
 a | b | c | d
---+---+---+---
 10 | 11 | -10 | -10
(1 строка)
```

Функция FLOOR

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

SELECT FLOOR(10) A, FLOOR(10.5) B, FLOOR(-10.5) C, FLOOR(-10) D;
D;

```
itco2=# SELECT FLOOR(10) A, FLOOR(10.5) B, FLOOR(-10.5) C, FLOOR(-10) D;  
 a | b | c | d  
---+---+---+---  
 10 | 10 | -11 | -10  
(1 строка)
```

Разница между FLOOR() и CEIL() в том, что в случае, когда FLOOR() возвращает ближайшее наименьшее, а CEIL() ближайшее наибольшее.

Функция TRUNC

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.

SELECT TRUNC(10.232443) A, TRUNC(-10.232443, 3) B,
TRUNC(10.99) C, TRUNC(10.25678, 2) D;

```
itco2=# SELECT TRUNC(10.232443) A  
 a | b | c | d  
---+---+---+---  
 10 | -10.232 | 10 | 10.25  
(1 строка)
```

Функция ROUND

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.

SELECT ROUND(10.232443) A, ROUND(10.5) B, ROUND(10.99) C,
ROUND(10.232443, 2) D;

```
itco2=# SELECT ROUND(10.232443)  
 a | b | c | d  
---+---+---+---  
 10 | 11 | 11 | 10.23  
(1 строка)
```

Функция SIGN

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

```
SELECT SIGN(10.22) A, SIGN(10) B, SIGN(-100) C, SIGN(0) D;
```

```
itco2=# SELECT SIGN(10
a | b | c | d
---+---+---+---
1 | 1 | -1 | 0
(1 строка)
```

Функция MOD

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

Например:

```
SELECT MOD(10, 3) A, MOD(10, 2) B, MOD(10, 98) C;
```

```
itco2=# SELECT MOD(10,
a | b | c
---+---+---
1 | 0 | 10
(1 строка)
```

Функция POWER

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

```
SELECT POWER(10, 2) A, POWER(100, 0.5) B, POWER(1000,
0.33333333) C, POWER(1000, -0.33333333) D;
```

```
itco2=# SELECT POWER(10, 2) A, POWER(100, 0.5) B, POWER(1000, 0.33333333
a | b | c | d
---+---+---+---
100 | 10.0000000000000000 | 9.9999997697414934 | 0.1000000023025851
(1 строка)
```

Функция SQRT

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

Например:

SELECT SQRT(16) A;

```
itco2=# SELECT SQRT(16) A;
 a
---
 4
(1 строка)
```

Функции EXP и LN

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).

Пример: SELECT EXP(1) A, LN(1) B, LN(EXP(2)) C;

```
itco2=# SELECT EXP(1) A, LN(1) B, LN(EXP(2)) C;
      a      | b | c
-----+---+---
2.718281828459045 | 0 | 2
(1 строка)
```

Функция LOG

Функция LOG(n, m) производит вычисление логарифма m по основанию n. Пример: SELECT LOG(2, 8) A, LOG(10, 100) B;

```
itco2=# SELECT LOG(2, 8) A, LOG(10, 100) B;
      a      | b
-----+-----
3.0000000000000000 | 2.0000000000000000
(1 строка)

itco2=# SELECT ROUND(LOG(2, 8), 0) A, ROUND(LOG(10, 100), 0) B;
 a | b
---+---
 3 | 2
(1 строка)
```

5.4 тригонометрические функции

- 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);

- $\text{COT}(n)$ – котангенс n (где n – угол в радианах).

Пример: `SELECT SIN(0) A, COS(0) B, TAN(0) C, COT(0) C;`

```
itco2=# SELECT SIN(0) A, COS(0) B, TAN(0) C, COT(0) C;
 a | b | c |      c
-----+-----+-----+-----
 0 | 1 | 0 | Infinity
(1 строка)
```

5.5 Строковые и символьные функции

Функция CONCAT

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`.

Пример:

`SELECT CONCAT ('пример ', 'для ', 'соединения ') A, CONCAT('пример', NULL) B, CONCAT(NULL, 'пример') C, CONCAT(NULL, NULL) D;`

```
itco2=# SELECT CONCAT('пример ', 'для ', 'соединения ') A,
CONCAT('пример', NULL);
      a          |      b      |      c      | concat
-----+-----+-----+-----
пример для соединения | пример      | пример      |
(1 строка)
```

Так же можно использовать оператор `||`:

`SELECT CONCAT('пример ', 'для ', 'соединения ') A, 'пример ' || 'для ' || 'соединения' B;`

```
itco2=# SELECT CONCAT('пример ', 'для ', 'соединения ') A,
'пример ' || 'для ' || 'соединения' B;
      a          |      b          |
-----+-----+-----
пример для соединения | пример для соединения
(1 строка)
```

Функция LOWER

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

```
itco2=# SELECT LOWER('ExAmPlE') A;
 a
-----
example
(1 строка)
```

Функция UPPER

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

SELECT UPPER('example') A;

```
itco2=# SELECT UPPER('example') A;
      а
-----
EXAMPLE
(1 строка)
```

Функция INITCAP

Функция INITCAP(str) возвращает строку str, в которой первые буквы всех слов преобразованы в прописные. Функция удобна для форматирования полного имени при построении отчетов.

Пример: SELECT INITCAP('Иванов петр сидорович') A;

```
itco2=# SELECT INITCAP('Иванов петр сидорович') A;
      а
-----
Иванов Петр Сидорович
(1 строка)
```

Функции LTRIM и RTRIM

Функция LTRIM(str [,set]) удаляет все символы с начала строки до первого символа, которого нет в наборе символов set. По умолчанию set состоит из одного пробела и может не указываться. Функция RTRIM(str [,set]) аналогична LTRIM, но удаляет символы, начиная от конца строки.

SELECT LTRIM('12345 TEST EXAMPLE', '12345 ') A, RTRIM('TEST EXAMPLE _#', '_# ') B;

```
itco2=# SELECT LTRIM('12345 TEST EXAMPLE', '12345 ') A, RTRIM('TEST EXAMPLE _#', '_# ') B;
      а      |      б
-----+-----
TEST EXAMPLE | TEST EXAMPLE
(1 строка)
```


Функция REPLACE

Функция REPLACE(str, search_str, replace_str) осуществляет поиск образца search_str в строке str и каждое найденное вхождение заменяет на replace_str. Поиск подстроки ведется с учетом регистра.

```
SELECT REPLACE('TEST EXAMPLE', 'E', '12345');
```

```
itco2=# SELECT REPLACE('TEST EXAMPLE', 'E', '12345');
         replace
-----
T12345ST 12345XAMPL12345
(1 строка)
```

Функция TRANSLATE

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. Для корректной работы функции строки from_mask и to_mask должны иметь одинаковую длину или строка from_mask должна быть длиннее, чем to_mask. Если from_mask длиннее, чем to_mask, и в процессе обработки строки str обнаружатся символы, соответствующие одному из символов from_mask, и при этом им не найдется соответствия в to_mask, то такие символы будут удалены из строки str. Если передать from_mask или to_mask значение, равное NULL, то функция возвратит значение NULL. Сравнение производится с учетом регистра.

```
SELECT TRANSLATE('TEST EXAMPLE', 'TE', '12');
```

```
itco2=# SELECT TRANSLATE('TEST EXAMPLE', 'TE', '12');
         translate
-----
12S1 2XAMPL2
(1 строка)
```

Функция SUBSTR

Функция SUBSTR(str, m) возвращает подстроку начиная с символа m.

```
itco2=# SELECT SUBSTR('TEST EXAMPLE', 3);
         substr
-----
ST EXAMPLE
(1 строка)
```

Функция LENGTH

Функция LENGTH(str) возвращает длину строки str в символах. Для пустой строки функция вернет 0, а для значения NULL – NULL.

SELECT LENGTH('test example') a, LENGTH("") b, LENGTH(NULL) c;

```
itco2=# SELECT LENGTH('test example') a, LENGTH('') b, LENGTH(NULL) c;
 a | b | c
---+---+---
 12 | 0 | 
(1 строка)
```

Функция ASCII

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

SELECT ASCII('example') a, ASCII('Example') b, ASCII("") c;

```
itco2=# SELECT ASCII('example') a, ASCII('Example') b, ASCII('') c;
 a | b | c
---+---+---
101 | 69 | 
(1 строка)
```

Функция CHR

Функция CHR(n) возвращает символ по его коду.

SELECT CHR(101) a, CHR(69) b;

```
itco2=# SELECT CHR(101) a, CHR(69) b;
 a | b
---+---
 e | E
(1 строка)
```

5.6 Функции работы с датой и временем

Функция NOW

Функция NOW возвращает текущую дату и время по часам сервера.

SELECT NOW();

```
itco2=# SELECT NOW();
 now
-----
2023-05-14 19:57:11.239909+03
(1 строка)
```

Функция AGE

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).

```
SELECT AGE(MAKE_DATE(2023, 7, 12), make_date(2021, 8,10));
```

```
itco2=# SELECT AGE(MAKE_DATE(2023, 7, 12), make_date(2021, 8, 10));
      age
-----
1 year 11 mons 2 days
(1 строка)
```

Функция EXTRACT

Функция EXTRACT(field FROM timestamp) извлекает элемент даты field из значения типа timestamp. Также существует функ- 5.6. Работа с датой и временем Глава 5. Дополнительные возможности ция EXTRACT(field FROM interval) для работы со значениями типа interval.

```
Select extract(month from now()) a, extract(year from now()) b,
extract(minute from now()) c;
```

```
itco2=# Select extract(month from now()) a, extract(year from now()) b,
extract(minute from now()) c;
 a |  b   |  c
---+-----+---
 5 | 2023 | 57
(1 строка)
```

Функция TO_DATE

Функция TO_DATE(str, mask) преобразует строку str в дату. Преобразование ведется по маске mask.

```
select to_date('01 JAN 20029', 'DD Mon YYYY');
```

```
itco2=# select to_date('01 JAN 20029', 'DD Mon YYYY');
 to_date
-----
20029-01-01
(1 строка)
```

Функция TO_CHAR

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

```
itco2=# select now() a, to_char(now(), 'YY.MM.DD HH24:MI:SS') b
          a                               |          b
-----+-----
 2023-05-16 18:14:44.084938+03 | 23.05.16 06:14
(1 строка)
```