МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

Факультет Управление и информатика в технологических системах
Кафедра Информационная безопасность
Специальность 10.05.03 «Информационная безопасность автоматизированных систем»

Дополнительные возможности POSTGRESQL

Выполнил студент гр. УБ-01
Цой Кирилл Вадимович

Воронеж – 2023

Функция ROW_NUMBER

Функция ROW_NUMBER генерирует порядковый номер строки запроса.

SELECT ROW_NUMBER() OVER (ORDER BY brand) Num, brand FROM auto;

```
CarSellingShop=# SELECT ROW_NUMBER() OVER (ORDER BY brand) Num, brand FROM auto;
 num | brand
-----+-------
   1 | CX-5
   2 | M8
   3 | RIO
   4 | RX-8
   5 | Soul
(5 строк)


CarSellingShop=#
```

Функция ABS

Функция ABS(n) возвращает абсолютное значение числа n.

SELECT ABS(13) X1, ABS(-300) X2, ABS(-100.2222) X3, ABS(1230.2222) X4;

```
CarSellingShop=# SELECT ABS(13) X1, ABS(-300) X2, ABS(-100.2222) X3, ABS(1230.2222) X4;
 x1 | x2  |    x3    |    x4
----+-----+----------+----------
 13 | 300 | 100.2222 | 1230.2222
(1 строка)


CarSellingShop=#
```

Функция CEIL

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

SELECT CEIL(13) X1, CEIL (-300) X2, CEIL (-100.2222) X3, CEIL (1230.2222) X4;

```
CarSellingShop=# SELECT CEIL(13) X1, CEIL (-300) X2, CEIL (-100.2222) X3, CEIL (1230.2222) X4;
 x1 |  x2  |  x3  |  x4
----+------+------+------
 13 | -300 | -100 | 1231
(1 строка)


CarSellingShop=#
```

Функция FLOOR

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

SELECT FLOOR (13) X1, FLOOR (-300) X2, FLOOR (-100.2222) X3, FLOOR (1230.2222) X4;

```
CarSellingShop=# SELECT FLOOR (13) X1, FLOOR (-300) X2, FLOOR (-100.2222) X3, FLOOR (1230.2222) X4;
 x1 |  x2  |  x3  |  x4
----+------+------+------
 13 | -300 | -101 | 1230
(1 строка)
```

Функция TRUNC

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

SELECT TRUNC (13) X1, TRUNC (-300) X2, TRUNC (-100.2222, 2) X3, TRUNC (1230.2222) X4;

```
CarSellingShop=# SELECT TRUNC (13) X1, TRUNC (-300) X2, TRUNC (-100.2222, 2) X3, TRUNC (1230.2222) X4;
 x1 |  x2  |   x3    |  x4
----+------+---------+------
 13 | -300 | -100.22 | 1230
(1 строка)


CarSellingShop=#
```

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 (13) X1, ROUND (-300) X2, ROUND (-100.2299, 2) X3, ROUND (1230.2222) X4;

```
CarSellingShop=# SELECT ROUND (13) X1, ROUND (-300) X2, ROUND (-100.2299, 2) X3, ROUND (1230.2222) X4;
 x1 |  x2  |   x3    |  x4
----+------+---------+------
 13 | -300 | -100.23 | 1230
(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 (13) X1, SIGN (-300) X2, SIGN (0) X3, SIGN (1230.2222) X4;

```
CarSellingShop=# SELECT SIGN (13) X1, SIGN (-300) X2, SIGN (0) X3, SIGN (1230.2222) X4;
 x1 | x2 | x3 | x4
----+----+----+----
  1 | -1 |  0 |  1
(1 строка)


CarSellingShop=#
```

Функция MOD

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

SELECT MOD (13,1) X1, MOD (-300,400) X2, MOD (33,2) X3, MOD (-1230, -4) X4;

```
CarSellingShop=# SELECT MOD (13,1) X1, MOD (-300,400) X2, MOD (33,2) X3, MOD (-1230, -4) X4;
 x1 |  x2  | x3 | x4
----+------+----+----
  0 | -300 |  1 | -2
(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) X1, POWER(100, 0.5) X2, POWER(1000, 0.33333333) X3, POWER(1000, -0.33333333) X4;

```
CarSellingShop=# SELECT POWER(10, 2) X1, POWER(100, 0.5) X2, POWER(1000, 0.33333333) X3, POWER(1000, -0.33333333) X4;
 x1  |        x2         |        x3         |        x4
-----+-------------------+-------------------+-------------------
 100 | 10.000000000000000 | 9.9999997697414934 | 0.1000000023025851
(1 строка)


CarSellingShop=# _
```

Функция SQRT

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

SELECT SQRT(100) X1, SQRT(25) X2, SQRT(298) X3;

```
CarSellingShop=# SELECT SQRT(100) X1, SQRT(25) X2, SQRT(298) X3;
 x1 | x2 |        x3
----+----+-------------------
 10 |  5 | 17.26267650163207
(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).

SELECT EXP(1) X1, LN(1) X2, LN(EXP(2)) X3;

```
CarSellingShop=# SELECT EXP(1) X1, LN(1) X2, LN(EXP(2)) X3;
        x1         | x2 | x3
-------------------+----+----
 2.718281828459045 |  0 |  2
(1 строка)


CarSellingShop=#
```

Функция LOG

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

SELECT LOG(2, 8) X1, LOG(10, 100) X2;

```
CarSellingShop=# SELECT LOG(2, 8) X1, LOG(10, 100) X2;
         x1         |         x2
--------------------+--------------------
 3.0000000000000000 | 2.0000000000000000
(1 строка)


CarSellingShop=#
```

Тригонометрические функции

PostgreSQL поддерживает вычисление основных тригономет-

рических функций:

1) SIN(n) – синус n (где n – угол в радианах);
2) COS(n) – косинус n (где n – угол в радианах);
3) TAN(n) – тангенс n (где n – угол в радианах);
4) COT(n) – котангенс n (где n – угол в радианах).

SELECT SIN(0) X1, COS(0) X2, TAN(0) X3, COT(0);

```
CarSellingShop=# SELECT SIN(0) X1, COS(0) X2, TAN(0) X3, COT(0);
 x1 | x2 | x3 |   cot
----+----+----+----------
  0 |  1 |  0 | Infinity
(1 строка)
```

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.

SELECT CONCAT('У', ' попа', ' была', ' собака') X1,

CONCAT('Test', NULL) X2,

CONCAT(NULL, 'Test') X3,

CONCAT(NULL, NULL) X4;

```
CarSellingShop=# SELECT CONCAT('У', ' попа', ' была', ' собака') X1,
CarSellingShop-# CONCAT('Test', NULL) X2,
CarSellingShop-# CONCAT(NULL, 'Test') X3,
CarSellingShop-# CONCAT(NULL, NULL) X4;
         x1          |  x2  |  x3  | x4
---------------------+------+------+----
 У попа была собака | Test | Test |
(1 строка)


CarSellingShop=#
```

Функция LOWER

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

SELECT LOWER('ИнФоРмАцИоНнАя БеЗоПаСнОсТь АвТоМаТиЗиРоВаНнЫх СиСтЕм') X;

```
CarSellingShop=# SELECT LOWER('ИнФоРмАцИоНнАя БеЗоПаСнОсТь АвТоМаТиЗиРоВаНнЫх СиСтЕм') X;
                          x
-------------------------------------------------------
 информационная безопасность автоматизированных систем
(1 строка)


CarSellingShop=#
```

Функция UPPER

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

SELECT UPPER ('ИнФоРмАцИоНнАя БеЗоПаСнОсТь АвТоМаТиЗиРоВаНнЫх СиСтЕм') X;

```
CarSellingShop=# SELECT UPPER ('ИнФоРмАцИоНнАя БеЗоПаСнОсТь АвТоМаТиЗиРоВаНнЫх СиСтЕм') X;
                          x
-------------------------------------------------------
 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ АВТОМАТИЗИРОВАННЫХ СИСТЕМ
(1 строка)


CarSellingShop=#
```

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.

SELECT INITCAP ('ИнФоРмАцИоНнАя БеЗоПаСнОсТь АвТоМаТиЗиРоВаНнЫх СиСтЕм') X;

```
CarSellingShop=# SELECT INITCAP ('ИнФоРмАцИоНнАя БеЗоПаСнОсТь АвТоМаТиЗиРоВаНнЫх СиСтЕм') X;
                          x
-----------------------------------------------------
 Информационная Безопасность Автоматизированных Систем
(1 строка)


CarSellingShop=#
```

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.

```
CarSellingShop=# SELECT LTRIM(' TeXt DATA') X1,
CarSellingShop-# LTRIM(' _ # TeXt DATA', ' #_') X2,
CarSellingShop-# LTRIM(' 1234567890 TeXt DATA', ' 1234567890') X3
CarSellingShop-# UNION ALL
CarSellingShop-# SELECT RTRIM('TeXt DATA ') X1,
CarSellingShop-# RTRIM('TeXt DATA _ # ', ' #_') X2,
CarSellingShop-# RTRIM('TeXt DATA 1234567890 ', ' 1234567890') X3;
    x1     |    x2     |    x3
-----------+-----------+-----------
 TeXt DATA | TeXt DATA | TeXt DATA
 TeXt DATA | TeXt DATA | TeXt DATA
(2 строки)


CarSellingShop=#
```

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.

```
CarSellingShop=# SELECT REPLACE('У попа была собака', 'собака', 'кошка') X1,
CarSellingShop-# REPLACE('У попа была собака', 'злая', '') X2,
CarSellingShop-# REPLACE('У попа была собака', 'Собака', 'Кошка') X3;
        x1         |         x2         |         x3
-------------------+--------------------+--------------------
 У попа была кошка | У попа была собака | У попа была собака
(1 строка)


CarSellingShop=#
```

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.

SELECT TRANSLATE('Test 12345', 'e2', 'E!') X1,

TRANSLATE('Test 12345', 'e234', 'E') X2;

```
CarSellingShop=# SELECT TRANSLATE('Test 12345', 'e2', 'E!') X1,
CarSellingShop-# TRANSLATE('Test 12345', 'e234', 'E') X2;
     x1     |   x2
------------+---------
 TEst 1!345 | TEst 15
(1 строка)


CarSellingShop=#
```

Функция SUBSTR

Функция SUBSTR(str, m [,n]) возвращает фрагмент строки str, начиная с символа m длиной n символов.

SELECT SUBSTR('У попа была собака', 13) X1,

SUBSTR('У попа была собака', -6) X2,

SUBSTR('Это тестовый текст', 5, 8) X3,

SUBSTR('У попа была собака', 150) X4;

```
CarSellingShop=# SELECT SUBSTR('У попа была собака', 13) X1,
CarSellingShop-# SUBSTR('У попа была собака', -6) X2,
CarSellingShop-# SUBSTR('Это тестовый текст', 5, 8) X3,
CarSellingShop-# SUBSTR('У попа была собака', 150) X4;
   x1   |         x2          |    x3    | x4
--------+---------------------+----------+----
 собака | У попа была собака | тестовый |
(1 строка)
```

Функция LENGTH

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

SELECT LENGTH('У попа была собака') X1,

LENGTH('') X2,

LENGTH(NULL) X3;

```
CarSellingShop=# SELECT LENGTH('У попа была собака') X1,
CarSellingShop-# LENGTH('') X2,
CarSellingShop-# LENGTH(NULL) X3;
 x1 | x2 | x3
----+----+----
 18 |  0 |
(1 строка)


CarSellingShop=#
```

Функция ASCII

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

SELECT ASCII('Test') X1, ASCII('Тест') X2;

```
CarSellingShop=# SELECT
CarSellingShop-# ASCII('Test') X1,
CarSellingShop-# ASCII('Тест') X2;
 x1 |  x2
----+------
 84 | 1058
(1 строка)


CarSellingShop=#
```

Функция CHR

Функция CHR(n) возвращает символ по его коду.

SELECT CHR(84) X1, CHR(1058) X2, CHR(80) X3, CHR(1060) X4;

```
CarSellingShop=# SELECT CHR(84) X1, CHR(1058) X2, CHR(80) X3, CHR(1060) X4;
 x1 | x2 | x3 | x4
----+----+----+----
 T  | Т  | P  | Ф
(1 строка)


CarSellingShop=#
```

Функция NOW

Возвращает текущую дату и время по часам сервера и часовой пояс.

SELECT NOW();

```
CarSellingShop=# SELECT NOW();
              now
-------------------------------
 2023-05-05 22:12:41.430486+03
(1 строка)


CarSellingShop=#
```

Функция JUSTIFY_INTERVAL

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

```
SELECT NOW() D1,
NOW() + JUSTIFY_INTERVAL('30 DAYS 1 HOUR 2 MINUTE') D2,
NOW() - JUSTIFY_INTERVAL('30 DAYS 1 HOUR 2 MINUTE') D3;
```

```
CarSellingShop=# SELECT NOW() D1,
CarSellingShop-# NOW() + JUSTIFY_INTERVAL('30 DAYS 1 HOUR 2 MINUTE') D2,
CarSellingShop-# NOW() - JUSTIFY_INTERVAL('30 DAYS 1 HOUR 2 MINUTE') D3;
              d1               |              d2               |              d3
-------------------------------+-------------------------------+-------------------------------
 2023-05-05 22:14:03.830322+03 | 2023-06-05 23:16:03.830322+03 | 2023-04-05 21:12:03.830322+03
(1 строка)


CarSellingShop=#
```

Функция DATE_TRUNC

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

```
SELECT
DATE_TRUNC('HOUR', NOW()) D1,
DATE_TRUNC('DAY', NOW()) D2,
DATE_TRUNC('MONTH', NOW()) D3;
```

```
CarSellingShop=# SELECT
CarSellingShop-# DATE_TRUNC('HOUR', NOW()) D1,
CarSellingShop-# DATE_TRUNC('DAY', NOW()) D2,
CarSellingShop-# DATE_TRUNC('MONTH', NOW()) D3;
           d1           |           d2           |           d3
------------------------+------------------------+------------------------
 2023-05-05 22:00:00+03 | 2023-05-05 00:00:00+03 | 2023-05-01 00:00:00+03
(1 строка)
```

Функция AGE

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

```
SELECT
CURRENT_DATE D1,
AGE(MAKE_TIMESTAMP(2013,  7, 15, 8, 15, 23.5)) D2,
AGE(MAKE_DATE(2016,  3, 3),
MAKE_TIMESTAMP(2013,  7, 15, 8, 15, 23.5)) D3;
```

```
CarSellingShop=# SELECT
CarSellingShop-# CURRENT_DATE D1,
CarSellingShop-# AGE(MAKE_TIMESTAMP(2013, 7, 15, 8, 15, 23.5)) D2,
CarSellingShop-# AGE(MAKE_DATE(2016, 3, 3),
CarSellingShop(# MAKE_TIMESTAMP(2013, 7, 15, 8, 15, 23.5)) D3;
     d1     |               d2                |               d3
------------+-----------------------------------+---------------------------------
 2023-05-05 | 9 years 9 mons 20 days 15:44:36.5 | 2 years 7 mons 18 days 15:44:36.5
(1 строка)
```

Функция EXTRACT

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

SELECT

NOW() D1,

EXTRACT(MONTH FROM NOW()) D2,

EXTRACT(YEAR FROM NOW()) D3,

EXTRACT(MINUTE FROM NOW()) D4;

```
              d1                | d2 |  d3  | d4
--------------------------------+----+------+----
 2023-05-05 22:24:06.453528+03 |  5 | 2023 | 24
(1 строка)


CarSellingShop=#
```

Функция TO_DATE

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

SELECT

TO_DATE('05 Dec 2000', 'DD Mon YYY') D1,

TO_DATE('15.12.2000', 'dd.mm.yy') D2;

```
CarSellingShop=# SELECT
CarSellingShop-# TO_DATE('05 Dec 2000', 'DD Mon YYY') D1,
CarSellingShop-# TO_DATE('15.12.2000', 'dd.mm.yy') D2;
     d1     |     d2
------------+------------
 2000-12-05 | 2000-12-15
(1 строка)


CarSellingShop=#
```

Функция TO_CHAR

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

SELECT NOW() D1,

TO_CHAR(NOW(), 'DD.MM.YY HH24:MI') D2;

```
CarSellingShop=# SELECT NOW() D1,
CarSellingShop-# TO_CHAR(NOW(), 'DD.MM.YY HH24:MI') D2;
              d1               |       d2
-------------------------------+----------------
 2023-05-05 22:25:35.173212+03 | 05.05.23 22:25
(1 строка)


CarSellingShop=#
```