# Лабораторная работа 20. Числовые функции MySQL

Цель: изучить числовые функции MySQL: MOD, FLOOR, ABS и т.п.

Ссылка: https://phpclub.ru/mysql/doc/arithmetic-functions.html

#### Задание. Выполнить в MySQL все последующие запросы. По возможности в качестве одного из аргументов использовать номер своего варианта.

#### Математические функции

Все математические функции в случае ошибки возвращают NULL.

* -

Унарный минус. Изменяет знак аргумента:

SELECT - 2;

-2

Необходимо учитывать, что если этот оператор используется с данными типа BIGINT, возвращаемое значение также будет иметь тип BIGINT! Это означает, что следует избегать использования оператора для целых чисел, которые могут иметь величину -2^63!

* ABS(X)

Возвращает абсолютное значение величины X:

SELECT ABS(2);

2

SELECT ABS(-32);

32

Эту функцию можно уверенно применять для величин типа BIGINT.

* SIGN(X)

Возвращает знак аргумента в виде -1, 0 или 1, в зависимости от того, является ли X отрицательным, нулем или положительным:

SELECT SIGN(-32);

-1

SELECT SIGN(0);

0

SELECT SIGN(234);

1

* MOD(N,M) , %

Значение по модулю (подобно оператору % в C). Возвращает остаток от деления N на M:

SELECT MOD(234, 10);

4

SELECT 253 % 7;

1

SELECT MOD(29,9);

2

Эту функцию можно уверенно применять для величин типа BIGINT.

* FLOOR(X)

Возвращает наибольшее целое число, не превышающее X:

SELECT FLOOR(1.23);

1

SELECT FLOOR(-1.23);

-2

Следует учитывать, что возвращаемая величина преобразуется в BIGINT!

* CEILING(X)

Возвращает наименьшее целое число, не меньшее, чем X:

SELECT CEILING(1.23);

2

SELECT CEILING(-1.23);

-1

Следует учитывать, что возвращаемая величина преобразуется в BIGINT!

* ROUND(X)

Возвращает аргумент X, округленный до ближайшего целого числа:

SELECT ROUND(-1.23);

-1

SELECT ROUND(-1.58);

-2

SELECT ROUND(1.58);

2

Следует учитывать, что поведение функции ROUND() при значении аргумента, равном середине между двумя целыми числами, зависит от конкретной реализации библиотеки C. Округление может выполняться: к ближайшему четному числу, всегда к ближайшему большему, всегда к ближайшему меньшему, всегда быть направленным к нулю. Чтобы округление всегда происходило только в одном направлении, необходимо использовать вместо данной хорошо определенные функции, такие как TRUNCATE() или FLOOR().

* ROUND(X,D)

Возвращает аргумент X, округленный до числа с D десятичными знаками. Если D равно 0, результат будет представлен без десятичного знака или дробной части:

SELECT ROUND(1.298, 1);

1.3

SELECT ROUND(1.298, 0);

1

* EXP(X)

Возвращает значение e (основа натуральных логарифмов), возведенное в степень X:

SELECT EXP(2);

7.389056

SELECT EXP(-2);

0.135335

* LN(X)

Возвращает натуральный логарифм числа X:

SELECT LN(2);

0.693147

SELECT LN(-2);

NULL

Эта функция появилась в MySQL 4.0.3. Это синоним LOG(X).

* LOG(B, X)

Если вызывается с одним параметром, возвращает натуральный логарифм числа X:

SELECT LOG(2);

0.693147

SELECT LOG(-2);

NULL

Если вызывается с двумя параметрами, функция возвращает логарифм числа X по базе B:

SELECT LOG(2,65536);

16.000000

SELECT LOG(1,100);

NULL

Опция указания базы логарифма появилась в MySQL 4.0.3. LOG(B,X) эквалиентно LOG(X)/LOG(B).

* LOG2(X)

Возвращает логарифм числа X по базе 2:

SELECT LOG2(65536);

16.000000

SELECT LOG2(-100);

NULL

Функция LOG2() полезна с тем, чтобы узнать, сколько бит число потребует для хранения. Эта функция добавлена в MySQL 4.0.3. В более старых версиях вызывайте вместо нее: LOG(X)/LOG(2)

* LOG10(X)

Возвращает десятичный логарифм числа X:

SELECT LOG10(2);

0.301030

SELECT LOG10(100);

2.000000

SELECT LOG10(-100);

NULL

* POW(X,Y) , POWER(X,Y)

Возвращает значение аргумента X, возведенное в степень Y:

SELECT POW(2,2);

4.000000

SELECT POW(2,-2);

0.250000

* SQRT(X)

Возвращает неотрицательный квадратный корень числа X:

SELECT SQRT(4);

2.000000

SELECT SQRT(20);

4.472136

* PI()

Возвращает значение числа "пи". По умолчанию представлено 5 десятичных знаков, но в MySQL для представления числа "пи" при внутренних вычислениях используется полная двойная точность.

SELECT PI();

3.141593

SELECT PI()+0.000000000000000000;

3.141592653589793116

* COS(X)

Возвращает косинус числа X, где X задается в радианах:

SELECT COS(PI());

-1.000000

* SIN(X)

Возвращает синус числа X, где X задается в радианах:

SELECT SIN(PI());

0.000000

* TAN(X)

Возвращает тангенс числа X, где X задается в радианах:

SELECT TAN(PI()+1);

1.557408

* ACOS(X)

Возвращает арккосинус числа X, т.е. величину, косинус которой равен X. Если X не находится в диапазоне от -1 до 1, возвращает NULL:

SELECT ACOS(1);

0.000000

SELECT ACOS(1.0001);

NULL

SELECT ACOS(0);

1.570796

* ASIN(X)

Возвращает арксинус числа X, т.е. величину, синус которой равен X. Если X не находится в диапазоне от -1 до 1, возвращает NULL:

SELECT ASIN(0.2);

0.201358

SELECT ASIN('foo');

0.000000

* ATAN(X)

Возвращает арктангенс числа X, т.е. величину, тангенс которой равен X:

SELECT ATAN(2);

1.107149

SELECT ATAN(-2);

-1.107149

* ATAN(Y,X) , ATAN2(Y,X)

Возвращает арктангенс двух переменных X и Y. Вычисление производится так же, как и вычисление арктангенса Y / X, за исключением того, что знаки обоих аргументов используются для определения квадранта результата:

SELECT ATAN(-2,2);

-0.785398

SELECT ATAN2(PI(),0);

1.570796

* COT(X)

Возвращает котангенс числа X:

SELECT COT(12);

-1.57267341

SELECT COT(0);

NULL

* RAND() , RAND(N)

Возвращает случайную величину с плавающей точкой в диапазоне от 0 до 1,0. Если целочисленный аргумент N указан, то он используется как начальное значение этой величины:

SELECT RAND();

0.9233482386203

SELECT RAND(20);

0.15888261251047

SELECT RAND(20);

0.15888261251047

SELECT RAND();

0.63553050033332

SELECT RAND();

0.70100469486881

В выражениях вида ORDER BY не следует использовать столбец с величинами RAND(), поскольку применение оператора ORDER BY приведет к многократным вычислениям в этом столбце. В версии MySQL 3.23 можно, однако, выполнить следующий оператор: SELECT \* FROM table\_name ORDER BY RAND(): он полезен для получения случайного экземпляра из множестваSELECT \* FROM table1,table2 WHERE a=b AND c<d ORDER BY RAND() LIMIT 1000. Следует учитывать, что оператор RAND() в выражении WHERE при выполнении выражения WHEREбудет вычисляться каждый раз заново. Оператор RAND() не следует воспринимать как полноценный генератор случайных чисел: это просто быстрый способ динамической генерации случайных чисел, переносимых между платформами для одной и той же версии MySQL.

* LEAST(X,Y,...)

Если задано два или более аргументов, возвращает наименьший (с минимальным значением) аргумент. Сравнение аргументов происходит по следующим правилам:

* + Если возвращаемая величина используется в целочисленном контексте (INTEGER), или все аргументы являются целочисленными, то они сравниваются как целые числа.
  + Если возвращаемая величина используется в контексте действительных чисел (REAL) или все аргументы являются действительными числами, то они сравниваются как числа типа REAL.
  + Если один из аргументов является зависимой от регистра строкой, то данные аргументы сравниваются с учетом регистра.
  + В остальных случаях аргументы сравниваются как строки, независимые от регистра.

SELECT LEAST(2,0);

0

SELECT LEAST(34.0,3.0,5.0,767.0);

3.0

SELECT LEAST("B","A","C");

"A"

В версиях MySQL до 3.22.5 можно использовать MIN() вместо LEAST.

* GREATEST(X,Y,...)

Возвращает наибольший (с максимальным значением) аргумент. Сравнение аргументов происходит по тем же правилам, что и для LEAST:

SELECT GREATEST(2,0);

2

SELECT GREATEST(34.0,3.0,5.0,767.0);

767.0

SELECT GREATEST("B","A","C");

"C"

В версиях MySQL до 3.22.5 можно использовать MAX() вместо GREATEST.

* DEGREES(X)

Возвращает аргумент X, преобразованный из радианов в градусы:

SELECT DEGREES(PI());

180.000000

* RADIANS(X)

Возвращает аргумент X, преобразованный из градусов в радианы:

SELECT RADIANS(90);

1.570796

* TRUNCATE(X,D)

Возвращает число X, усеченное до D десятичных знаков. Если D равно 0, результат будет представлен без десятичного знака или дробной части:

SELECT TRUNCATE(1.223,1);

1.2

SELECT TRUNCATE(1.999,1);

1.9

SELECT TRUNCATE(1.999,0);

1

SELECT TRUNCATE(-1.999,1);

-1.9

Начиная с MySQL 3.23.51 все числа округляются к нулю.

Если D - негативное, то тогда вся часть числа обнуляется:

SELECT TRUNCATE(122,-2);

100

Следует учитывать, что обычно в компьютерах десятичные числа хранятся не так, как целые, а как числа двойной точности с плавающим десятичным знаком (DOUBLE). Поэтому иногда результат может вводить в заблуждение, как в следующем примере:

SELECT TRUNCATE(10.28\*100,0);

1027

Это происходит потому, что в действительности 10,28 хранится как нечто вроде 10,2799999999999999.

**Контрольные вопросы.**

1. Математические функции, использующие один аргумент

2. Математические функции, использующие два аргумента.

3. Математические функции, использующие более двух аргументов.