

MySQL

Операторы и функции

Арифметические операции

Пример	Название
$-a$	Отрицание
$a + b$	Сложение
$a - b$	Вычитание
$a * b$	Умножение
a / b	Деление

Битовые операции

Пример	Название
$a \& b$	Побитовое 'И'
$a b$	Побитовое 'ИЛИ'
$\sim a$	Отрицание
$a \ll b$	Сдвиг влево
$a \gg b$	Сдвиг вправо
<code>BIT_COUNT(N)</code>	Подсчет

Операторы сравнения

Оператор	Название
$a = b$	Равно
$a < > b, a \neq b$	Не равно
$a < b$	Меньше
$a > b$	Больше
$a \leq b$	Меньше или равно
$a \geq b$	Больше или равно
$a \langle \Rightarrow \rangle b$	NULL безопасное сравнение

Операторы сравнения

Оператор	Название
a IS NULL, ISNULL(a)	Определение на NULL
a BETWEEN b AND c	Вхождение в диапазон
a IN(b, c, d)	Вхождение в список
COALESCE(a, b, c)	Поиск первого не NULL значения списка
INTERVAL(a, b, c, d)	Отношение к интервалу

Логические операторы

Оператор	Название
NOT	Логическое «НЕ»
AND	Логическое «И»
OR	Логическое «ИЛИ»

Функции условий

■ IFNULL(a, b)

- `SELECT IFNULL(1, 0);` // вернет 1
- `SELECT IFNULL(NULL, 10);` // вернет 10
- `SELECT IFNULL(NULL, 'yes');` // вернет 'yes'

■ NULLIF(a, b)

- `SELECT NULLIF(1, 1);` // вернет NULL
- `SELECT NULLIF(1,2);` // вернет 1

■ IF(a, b, c)

- `SELECT IF(1>2,2,3 `);` // вернет 3
- `SELECT IF(1<2,'yes','no');` // вернет 'yes'

Оператор поливариантного выбора

- CASE val
 - WHEN [compare] THEN result
 - [WHEN [compare] THEN result ...]
 - [ELSE result]
- END
- CASE WHEN [condition]
 - THEN result [WHEN [condition]
 - THEN result ...]
 - [ELSE result]
- END

Пример

- SELECT CASE 1
 - WHEN 1 THEN "one"
 - WHEN 2 THEN "two"
 - ELSE "more"
- END;
- SELECT CASE WHEN 1>0
 - THEN "true"
 - ELSE "false"
- END;

Поиск по образцу

- a LIKE маск;

- Маски

- % – любое число символов
- _ – один символ

- Примеры

- `SELECT 'Hello my friend!' LIKE '%friend';`
- `SELECT 'Hello my friend!' LIKE 'Hello%fr_end!';`
- `SELECT 'Hello my friend!' LIKE 'Hello_friend';`

Регулярные выражения

- a REGEXP masc

- a RLIKE masc

- *SELECT 'Monty!' REGEXP 'M%y%';*
- *SELECT 'Monty!' REGEXP '.*';*
- *SELECT 'new*\n*line' REGEXP 'new\\|*\\.\\|*line';*
- *SELECT "bri*" REGEXP "[a-d]r*[A-j]*";*

Сравнение строк

■ STRCMP(a, b);

- *SELECT STRCMP('text', 'text2');* // -1
- *SELECT STRCMP('text2', 'text');* // 1
- *SELECT STRCMP('text', 'text');* // 0

Реливантный поиск

- `MATCH(col1, col2, ...) AGAINST (expr);`
- `SELECT`
 - `* FROM articles`
 - `WHERE MATCH (title,body)`
 - `AGAINST ('database');`
- `SELECT`
 - `id,MATCH (title,body) AGAINST ('Tutorial')`
 - `FROM articles;`

Операторы полнотекстового поиска

Тип	Описание	Пример
+	Слово присутствует в строке	+apple +juice
-	Слово отсутствует в строке	+apple -mac
<, >	Изменение приоритета релевантности	+apple +(>pie <juice)
()	Подвыражение	+apple +(>pie <juice)
~	Негативный вклад слова	+apple ~juice
*	Усечение (ставится в конце)	apple*
""	Буквальный поиск	"apple juice"

Функции работы с датой

Функция	Описание
DAYOFWEEK(date), WEEKDAY(date)	Возвращает номер дня недели (вс, пн)
DAYOFMONTH(date), DAYOFYEAR(date)	Возвращает день месяца, день года
MONTH(date), YEAR(date)	Возвращает месяц, год для даты
MONTHNAME(date), DAYNAME(date)	Название месяца, дня недели
QUARTER(date)	Возвращает четверть года

Функции работы с датой

Функция	Описание
WEEK(date), WEEK(date,first)	Возвращает номер недели
YEARWEEK(date), YEARWEEK(date,first)	Возвращает год и неделю
HOUR(time), MINUTE(time), SECOND(time)	Возвращает часы, минуты, секунды
PERIOD_ADD(P,N)	Добавляет N месяцев к периоду P
PERIOD_DIFF(P1,P2)	Возвращает число месяцев между двумя периодами

Функции работы с датой

Функция	Описание
DATE_FORMAT(date,format)	Функция формируют дату по маске
TIME_FORMAT(time,format)	Функция формируют время по маске
CURDATE(), CURRENT_DATE()	Возвращает текущую дату (YYYY-MM-DD)
CURTIME(), CURRENT_TIME()	Возвращает текущее время (HH:MM:SS)

Функции работы с датой

Функция	Описание
UNIX_TIMESTAMP(), UNIX_TIMESTAMP(date)	Возвращает UNIX TIMESTAMP
FROM_UNIXTIME(uts,format)	Возвращает дату по UNIX TIMESTAMP
SEC_TO_TIME(seconds)	Получает время по количеству секунд
TIME_TO_SEC(time)	Получает количество секунд по дате

COUNT

■ COUNT(expr)

- SELECT `group_num`, COUNT(*)
- FROM `student`
- GROUP BY `group_num`;

AVG

■ AVG(expr)

- SELECT `group_num`, AVG(`test_score`) AS `avg`
- FROM `student` GROUP BY `group_num`
- HAVING `avg` > 50;

MAX, MIN

- **MIN(expr)**
- **MAX(expr)**
 - SELECT
 - `group_num`,
 - MIN(`test_score`) AS `min`,
 - MAX(`test_score`) AS `max`
 - FROM `student`
 - GROUP BY `group_num`;

SUM

- **SUM(expr)**
 - SELECT
 - SUM(`id_photo`) AS `sum`
 - FROM `album`