

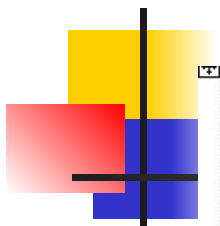
Лекция №3

Выборка данных (*SELECT*)

План лекции:

1. Синтаксис оператора *SELECT*.
2. Список выбора.
3. Инструкция *WHERE*.
4. Арифметические и логические операторы языка СУБД *MySQL*.
5. Использование встроенных функций СУБД *MySQL*:
 - агрегирующие функции;
 - математические функции;
 - строковые функции;
 - функции даты и времени.

Синтаксис оператора *SELECT*



• Синтаксис оператора <i>SELECT</i>	• Описание
<ul style="list-style-type: none"> • <code>SELECT</code> · [<code>DISTINCT</code> · · <code>ALL</code>] • [<code>{select_expr}</code> · [<code>[[AS]</code> · <code>alias</code>] • [<code>,</code> <code>select_expr</code> · ... ·] • <code>FROM</code> · <code>table_references</code> • [<code>WHERE</code> · <code>where_condition</code>] • [<code>GROUP</code> · <code>BY</code> · <code>{unsigned_integer</code> · • · · <code>col_name</code> · · <code>formula</code>}] • [<code>HAVING</code> · <code>where_condition</code>] • [<code>ORDER</code> · <code>BY</code> · <code>{unsigned_integer</code> · • · • <code>col_name</code> · } · [<code>ASC</code> · · <code>DESC</code>], · • ... ·] • [<code>LIMIT</code> · <code>{start</code>, · <code>amount</code> • · · <code>amount</code>}] 	<ul style="list-style-type: none"> • <code>DISTINCT</code> — только уникальные записи (без повторов); <code>ALL</code> — все записи; • <code>select_expr</code> — поле для выборки; • <code>FROM</code> — источник запроса (таблица, перечень таблиц, представления, подзапросы); • <code>WHERE</code> — задание условий для отбора записей; • <code>GROUP BY</code> — группировка данных результата запроса <code>SELECT</code> · ... · <code>WHERE</code> · ... ·; • <code>HAVING</code> — задание условий для отбора записей инструкцией <code>GROUP BY</code>; • <code>ORDER BY</code> — сортировка результата запроса (по номерам полей по именам полей). • <code>DESC</code> — сортировка по убыванию; • <code>LIMIT</code> — ограничение результатов запроса (номеру начальной записи и их количеству либо только по количеству записей)



Список выбора (select_expr)

Select_expr могут задаваться в виде :

- *col_name*;
- *tbl_name.col_name*;
- *db_name.tbl_name.col_name*.

(Если имя базы данных, таблицы или поля относится к служебным словам MySQL или содержит специальные символы, необходимо обязательно заключать его в обратные кавычки «` `», например:
`db_name`.`tbl_name`.`col_name`.)

Присвоение псевдонимов (AS)

col_name AS new_name.

(Синтаксис языка SQL разрешает не указывать оператор AS (не рекомендуется).

Указанные псевдонимы используются в предложении *where_condition*. В предложении *where_condition* можно указывать псевдонимы выражений *db_name*, *tbl_name*, *col_name*, только если эти ссылки можно истолковать однозначно.



Условия выборки (WHERE)

Для выбора записей, удовлетворяющих критериям поиска, используется инструкция WHERE и выражение where_condition. Выражение where_condition определяет условие (или условия), которым должны соответствовать отбираемые записи.

Это выражение принимает значение FALSE или TRUE для каждой записи, отбираемой из источника запроса.

Условие для отбора записи – where_condition = true.

В выражении WHERE могут использоваться операторы (арифметические, логические) и функции, за исключением агрегатных (суммирующих) функций.

Арифметические операторы СУБД MySQL

Оператор	Описание
=	Оператор равенства. Возвращает 1 (true), 1 (истина), если операнды равны, и 0 (false), если не равны
<=>	Оператор эквивалентности. <u>Аналогичен</u> обычному равенству, но возвращает только два значения: 1 (true), и 0 (false). NULL не возвращает
<>	Оператор неравенства. Возвращает 1 (true), если операнды не равны, и 0 (false), если равны
<	Оператор «меньше». Возвращает 1 (true), если левый операнд меньше правого, и 0 (false) — в противном случае
<=	Оператор «меньше или равно». Возвращает 1 (true), если левый операнд меньше правого или они равны, и 0 (false) — в противном случае
>	Оператор «больше». Возвращает 1 (true), если левый операнд больше правого, и 0 (false) — в противном случае
>=	Оператор «больше или равно». Возвращает 1 (true), если левый операнд больше правого или они равны, и 0 (false) — в противном случае

Логические операторы языка SQL

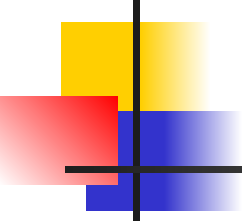
(TRUE=1, FALSE=0)

Оператор	Пример	Значение					
AND	n • AND • m	Логическое AND (логическое умножение).					
		n	m	n • AND • m		n	m
		FALSE	FALSE	FALSE		NULL	FALSE
		FALSE	TRUE	FALSE		NULL	TRUE
		TRUE	FALSE	FALSE		FALSE	NULL
		TRUE	TRUE	TRUE		TRUE	NULL
OR	n • OR • m	Логическое OR (логическое сложение).					
		n	m	n • OR • m		n	m
		FALSE	FALSE	FALSE		NULL	FALSE
		FALSE	TRUE	TRUE		NULL	TRUE
		TRUE	FALSE	TRUE		FALSE	NULL
		TRUE	TRUE	TRUE		TRUE	NULL

Логические операторы языка SQL (продолжение)

XOR	$n \cdot \text{XOR} \cdot m$	<div><div><div>⊕</div><div><div>Логическое XOR (исключающее OR)</div><table><tr><td>n</td><td>m</td><td>$n \cdot \text{XOR} \cdot m$</td><td></td><td>n</td><td>m</td><td>$n \cdot \text{XOR} \cdot m$</td></tr><tr><td>FALSE</td><td>FALSE</td><td>FALSE</td><td></td><td>NULL</td><td>FALSE</td><td>NULL</td></tr><tr><td>FALSE</td><td>TRUE</td><td>TRUE</td><td></td><td>NULL</td><td>TRUE</td><td>NULL</td></tr><tr><td>TRUE</td><td>FALSE</td><td>TRUE</td><td></td><td>FALSE</td><td>NULL</td><td>NULL</td></tr><tr><td>TRUE</td><td>TRUE</td><td>FALSE</td><td></td><td>TRUE</td><td>NULL</td><td>NULL</td></tr><tr><td></td><td></td><td></td><td></td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table><div>○</div><div>□</div></div></div></div>	n	m	$n \cdot \text{XOR} \cdot m$		n	m	$n \cdot \text{XOR} \cdot m$	FALSE	FALSE	FALSE		NULL	FALSE	NULL	FALSE	TRUE	TRUE		NULL	TRUE	NULL	TRUE	FALSE	TRUE		FALSE	NULL	NULL	TRUE	TRUE	FALSE		TRUE	NULL	NULL					NULL	NULL	NULL
n	m	$n \cdot \text{XOR} \cdot m$		n	m	$n \cdot \text{XOR} \cdot m$																																						
FALSE	FALSE	FALSE		NULL	FALSE	NULL																																						
FALSE	TRUE	TRUE		NULL	TRUE	NULL																																						
TRUE	FALSE	TRUE		FALSE	NULL	NULL																																						
TRUE	TRUE	FALSE		TRUE	NULL	NULL																																						
				NULL	NULL	NULL																																						
NOT	$\text{NOT} \cdot n$	<div><div><div>Логическое NOT (логическое отрицание, инверсия)</div><table><tr><td>n</td><td>$\text{NOT} \cdot n$</td></tr><tr><td>FALSE</td><td>TRUE</td></tr><tr><td>TRUE</td><td>FALSE</td></tr><tr><td>NULL</td><td>NULL</td></tr></table><div>○</div></div></div>	n	$\text{NOT} \cdot n$	FALSE	TRUE	TRUE	FALSE	NULL	NULL																																		
n	$\text{NOT} \cdot n$																																											
FALSE	TRUE																																											
TRUE	FALSE																																											
NULL	NULL																																											

Использование встроенных функций СУБД MySQL



Для решения специфических задач используют встроенные функции СУБД MySQL, которые могут использоваться:

- в выражениях `select_expr` оператора `SELECT`;
- выражении `where_condition` инструкций `WHERE` и `HAVING`.

Основные функции СУБД MySQL:

- агрегирующие функции;
- математические функции;
- строковые функции;
- функции даты и времени;

Агрегирующие функции MySQL

Обозначение	Описание
<code>AVG([DISTINCT] expr)</code>	Возвращает среднее значение аргумента <code>expr</code> . В качестве аргумента обычно используется имя поля записи. Необязательное слово <code>DISTINCT</code> позволяет обрабатывать только уникальные значения поля <code>expr</code> .
<code>COUNT(expr)</code> <code>COUNT(*)</code>	Подсчитывает число записей и имеет несколько форм. Форма <code>COUNT(expr)</code> возвращает число записей в таблице с полем <code>expr</code> не равных <code>NULL</code> . Форма <code>COUNT(*)</code> возвращает общее число записей в таблице независимо от того, принимает какое-либо поле значение <code>NULL</code> или нет. Форма <code>COUNT(DISTINCT expr1, expr2, ...)</code> позволяет подсчитать только уникальные значения полей.
<code>GROUP_CONCAT(·)</code>	Функция <code>GROUP_CONCAT(·)</code> объединяет в одном поле несколько полей запроса <code>SELECT</code> , вставляя между ними <i>разделитель</i> (по умолчанию — запятая). Синтаксис: <code>SELECT GROUP_CONCAT(col_name SEPARATOR разделитель) FROM tbl_name [WHERE · GROUP BY]</code>
<code>MIN([DISTINCT] expr)</code>	Возвращает минимальное значение среди всех <code>NOT NULL</code> значений выбранных записей по полю <code>expr</code> . Необязательное слово <code>DISTINCT</code> позволяет обрабатывать только уникальные значения поля <code>expr</code> .
<code>MAX([DISTINCT] expr)</code>	Возвращает максимальное значение среди всех <code>NOT NULL</code> значений выбранных записей по полю <code>expr</code> . Необязательное слово <code>DISTINCT</code> позволяет обрабатывать только уникальные значения поля <code>expr</code> .
<code>STD(expr)</code>	Возвращает стандартное среднеквадратичное отклонение в аргументе <code>expr</code> .
<code>STDDEV_SAMP(expr)</code>	Возвращает выборочное среднеквадратичное отклонение в аргументе <code>expr</code> .
<code>SUM([DISTINCT] expr)</code>	Возвращает сумму значений по полю <code>expr</code> . Необязательная инструкция <code>DISTINCT</code> позволяет учитывать только уникальные значения поля <code>expr</code> .

Математические функции MySQL

Синтаксис	Описание
$ABS \cdot (X)$	Возвращает абсолютное значение аргумента X
$CEIL \cdot (X)$	Возвращает наименьшее целочисленное значение, которое больше или равно числу X ($CEIL \cdot (X) \geq X$). Пример: $CEIL \cdot (12.34) = 13$
$FLOOR \cdot (X)$	Возвращает наибольшее целочисленное значение, которое меньше или равно числу X ($FLOOR \cdot (X) \leq X$). Пример: $FLOOR \cdot (12.34) = 12$
$MOD \cdot (M, N)$	Возвращает остаток от деления целого числа M на целое число N
$POW \cdot (X, Y)$	Возвращает значение числа X , возведенного в степень Y ($X^Y = POW(X, Y)$)
$RAND \cdot (X)$	Возвращает случайное значение с плавающей точкой (от 0.0 до 1.0)
$ROUND \cdot (X, D)$	Возвращает число X , округленное до определенного D количества знаков после запятой ($D=0, 1, 2 \dots$)
$SIGN \cdot (X)$	Определяет знак числа X . Если X отрицательно — возвращает минус один (-1), если положительно — единицу (1), если ноль — возвращает ноль (0)
$SQRT \cdot (X)$	Вычисляет квадратный корень числа X
$TRUNCATE \cdot (X, D)$ $D=0, 1, \dots$	Возвращает число X , усеченное до определенного D количества знаков после запятой ($D=0, 1, 2 \dots$). Если количество знаков в X больше D , лишние разряды игнорируются. Если меньше, то в конец числа добавляются нули (0).

Строковые функции MySQL

Синтаксис	Описание
ASCII·(str)	Возвращает значение ASCII-кода первого символа строки str. Для пустой строки возвращается значение ноль (0).
BIN·(N)	Принимает десятичное число N и возвращает его двоичное представление
BIT_LENGTH·(str)	Принимает строку str и возвращает ее длину в битах
CHAR·(N1,·N2,·...·)	Принимает последовательность из ASCII-кодов и возвращает строку, построенную путем объединения соответствующих им символов
CHAR_LENGTH·(str)	Принимает строку str и возвращает число символов в строке
CHARSET·(str)	Возвращает имя кодировки, в которой представлена строка
COLLATION·(str)	Возвращает порядок сортировки, установленный для кодировки аргумента str
CONCAT·(str1,·str2,·...·)	Возвращает строку, созданную путем объединения всех аргументов, количество которых не ограничено. Если хотя бы один аргумент равен NULL, то возвращается значение NULL
CONCAT_WS·(separator,·str1,·str2,·...·)	Также объединяет аргументы str1, str2 и т.д., помещая между ними разделитель separator. Игнорирует NULL.
CONV·(N,·from_base,·to_base)	Преобразует число N из одной системы счисления from_base в другую to_base. Параметры from_base и to_base могут принимать значения от 2 до 36
ELT·(N,·str1,·str2,·...·)	Возвращает N-ю строку из списка аргументов str1, str2, ... (для N=1 возвращается str1, для N=2 — str2 и т.д.)
FIELD·(str,·str1,·str2,·...·)	Находит строку str в списке str1, str2, ... и возвращает номер строки в этом списке (нумерация начинается с единицы (1))

Строковые функции MySQL *продолжение*

FIND_IN_SET(str, str_list)	Ищет вхождение строки str в список str_list и возвращает номер строки в этом списке (нумерация начинается с 1). Параметр str_list — набор строк, разделенных запятыми
INSERT(str, pos, len, new_str)	Возвращает строку str, в которой подстрока, начинающаяся с позиции pos и имеющая длину len символов, заменена подстрокой new_str
INSTR(str, substr)	Возвращает позицию первого вхождения подстроки substr в строку str
LEFT(str, len)	Возвращает len крайних левых символов строки str
LENGTH(str)	Возвращает длину строки str
LOCATE(substr, str[, pos])	Возвращает позицию первого вхождения подстроки substr в строку str. При наличии необязательного аргумента pos, поиск начинается с позиции, указанной в этом аргументе (pos)
LOWER(str)	Возвращает строку str, записанную строчными символами

Функции даты и времени MySQL

Синтаксис	Описание
ADDDATE·(date,·INTERVAL· expr·type)	Возвращает дату date, к которой прибавлен временной интервал, определяемый вторым параметром. Например, ADDDATE('2009-03-20',·INTERVAL·10·DAY)
ADDTIME·(expr1,·expr2)	Возвращает результат сложения двух временных значений
CURDATE·(·)	Возвращает текущую дату в формате 'YYYY-MM-DD'
CURTIME·(·)	Возвращает текущее время суток в формате 'hh:mm:ss'
DATE·(datetime)	Извлекает из значения datetime дату, отсекая часы, минуты и секунды
DATEDIFF·(begin,·end)	Вычисляет разницу в днях между датами begin и end
DATE_FORMAT·(date,· format)	Форматирует дату date в соответствии со строкой format
DAY·(date)	Возвращает порядковый номер дня в месяце (от 1 до 31)
DAYNAME·(date)	Возвращает день недели в виде полного английского названия
DAYOFWEEK·(date)	Возвращает порядковый номер дня недели. В западных странах неделя начинается с воскресенья, номер которого 1.
DAYOFYEAR·(date)	Возвращает порядковый номер дня в году (от 1 до 366)
EXTRACT·(type·FROM· datetime)	Принимает дату и время суток и возвращает часть, определяемую параметром type. Например, EXTRACT(YEAR·FROM·'2009-12-31·14:30:15')
FROM_DAYS·(N)	Принимает число дней N, прошедших с нулевого года, и возвращает дату в формате 'YYYY-MM-DD'. Обычно используется совместно с функцией TO_DAYS(date)
HOUR·(datetime)	Извлекает из значения datetime часы (от 0 до 23)
LAST_DAY·(datetime)	Принимает дату и время суток и возвращает дату — последний день текущего месяца