

# Лекция №3

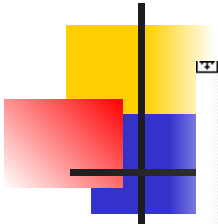
## Выборка данных (*SELECT*)

---

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



# 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	$n \cdot \text{AND} \cdot m$	<p>Логическое AND (логическое умножение).</p> <table border="1"> <thead> <tr> <th>n</th> <th>m</th> <th><math>n \cdot \text{AND} \cdot m</math></th> <th>n</th> <th>m</th> <th><math>n \cdot \text{AND} \cdot m</math></th> </tr> </thead> <tbody> <tr> <td>FALSE</td> <td>FALSE</td> <td>FALSE</td> <td>NULL</td> <td>FALSE</td> <td>FALSE</td> </tr> <tr> <td>FALSE</td> <td>TRUE</td> <td>FALSE</td> <td>NULL</td> <td>TRUE</td> <td>NULL</td> </tr> <tr> <td>TRUE</td> <td>FALSE</td> <td>FALSE</td> <td>FALSE</td> <td>NULL</td> <td>FALSE</td> </tr> <tr> <td>TRUE</td> <td>TRUE</td> <td>TRUE</td> <td>TRUE</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table>	n	m	$n \cdot \text{AND} \cdot m$	n	m	$n \cdot \text{AND} \cdot m$	FALSE	FALSE	FALSE	NULL	FALSE	FALSE	FALSE	TRUE	FALSE	NULL	TRUE	NULL	TRUE	FALSE	FALSE	FALSE	NULL	FALSE	TRUE	TRUE	TRUE	TRUE	NULL	NULL				NULL	NULL	NULL
n	m	$n \cdot \text{AND} \cdot m$	n	m	$n \cdot \text{AND} \cdot m$																																	
FALSE	FALSE	FALSE	NULL	FALSE	FALSE																																	
FALSE	TRUE	FALSE	NULL	TRUE	NULL																																	
TRUE	FALSE	FALSE	FALSE	NULL	FALSE																																	
TRUE	TRUE	TRUE	TRUE	NULL	NULL																																	
			NULL	NULL	NULL																																	
OR	$n \cdot \text{OR} \cdot m$	<p>Логическое OR (логическое сложение).</p> <table border="1"> <thead> <tr> <th>n</th> <th>m</th> <th><math>n \cdot \text{OR} \cdot m</math></th> <th>n</th> <th>m</th> <th><math>n \cdot \text{OR} \cdot m</math></th> </tr> </thead> <tbody> <tr> <td>FALSE</td> <td>FALSE</td> <td>FALSE</td> <td>NULL</td> <td>FALSE</td> <td>NULL</td> </tr> <tr> <td>FALSE</td> <td>TRUE</td> <td>TRUE</td> <td>NULL</td> <td>TRUE</td> <td>TRUE</td> </tr> <tr> <td>TRUE</td> <td>FALSE</td> <td>TRUE</td> <td>FALSE</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>TRUE</td> <td>TRUE</td> <td>TRUE</td> <td>TRUE</td> <td>NULL</td> <td>TRUE</td> </tr> <tr> <td></td> <td></td> <td></td> <td>NULL</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table>	n	m	$n \cdot \text{OR} \cdot m$	n	m	$n \cdot \text{OR} \cdot m$	FALSE	FALSE	FALSE	NULL	FALSE	NULL	FALSE	TRUE	TRUE	NULL	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	NULL	NULL	TRUE	TRUE	TRUE	TRUE	NULL	TRUE				NULL	NULL	NULL
n	m	$n \cdot \text{OR} \cdot m$	n	m	$n \cdot \text{OR} \cdot m$																																	
FALSE	FALSE	FALSE	NULL	FALSE	NULL																																	
FALSE	TRUE	TRUE	NULL	TRUE	TRUE																																	
TRUE	FALSE	TRUE	FALSE	NULL	NULL																																	
TRUE	TRUE	TRUE	TRUE	NULL	TRUE																																	
			NULL	NULL	NULL																																	

# Логические операторы языка SQL (продолжение)

XOR	$n \cdot \text{XOR} \cdot m$	<p data-bbox="919 444 1682 486">Логическое XOR (исключающее OR)</p> <table border="1" data-bbox="770 505 1827 819"> <thead> <tr> <th data-bbox="774 508 929 558">n</th> <th data-bbox="929 508 1083 558">m</th> <th data-bbox="1083 508 1277 558"><math>n \cdot \text{XOR} \cdot m</math></th> <th data-bbox="1315 508 1470 558">n</th> <th data-bbox="1470 508 1624 558">m</th> <th data-bbox="1624 508 1823 558"><math>n \cdot \text{XOR} \cdot m</math></th> </tr> </thead> <tbody> <tr> <td data-bbox="774 558 929 608">FALSE</td> <td data-bbox="929 558 1083 608">FALSE</td> <td data-bbox="1083 558 1277 608">FALSE</td> <td data-bbox="1315 558 1470 608">NULL</td> <td data-bbox="1470 558 1624 608">FALSE</td> <td data-bbox="1624 558 1823 608">NULL</td> </tr> <tr> <td data-bbox="774 608 929 658">FALSE</td> <td data-bbox="929 608 1083 658">TRUE</td> <td data-bbox="1083 608 1277 658">TRUE</td> <td data-bbox="1315 608 1470 658">NULL</td> <td data-bbox="1470 608 1624 658">TRUE</td> <td data-bbox="1624 608 1823 658">NULL</td> </tr> <tr> <td data-bbox="774 658 929 708">TRUE</td> <td data-bbox="929 658 1083 708">FALSE</td> <td data-bbox="1083 658 1277 708">TRUE</td> <td data-bbox="1315 658 1470 708">FALSE</td> <td data-bbox="1470 658 1624 708">NULL</td> <td data-bbox="1624 658 1823 708">NULL</td> </tr> <tr> <td data-bbox="774 708 929 758">TRUE</td> <td data-bbox="929 708 1083 758">TRUE</td> <td data-bbox="1083 708 1277 758">FALSE</td> <td data-bbox="1315 708 1470 758">TRUE</td> <td data-bbox="1470 708 1624 758">NULL</td> <td data-bbox="1624 708 1823 758">NULL</td> </tr> <tr> <td data-bbox="774 758 929 819"></td> <td data-bbox="929 758 1083 819"></td> <td data-bbox="1083 758 1277 819"></td> <td data-bbox="1315 758 1470 819">NULL</td> <td data-bbox="1470 758 1624 819">NULL</td> <td data-bbox="1624 758 1823 819">NULL</td> </tr> </tbody> </table>	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	$\text{NOT} \cdot n$	<p data-bbox="755 886 1846 929">Логическое NOT (логическое отрицание, инверсия)</p> <table border="1" data-bbox="1141 948 1456 1158"> <thead> <tr> <th data-bbox="1145 951 1300 1001">n</th> <th data-bbox="1300 951 1456 1001"><math>\text{NOT} \cdot n</math></th> </tr> </thead> <tbody> <tr> <td data-bbox="1145 1001 1300 1051">FALSE</td> <td data-bbox="1300 1001 1456 1051">TRUE</td> </tr> <tr> <td data-bbox="1145 1051 1300 1100">TRUE</td> <td data-bbox="1300 1051 1456 1100">FALSE</td> </tr> <tr> <td data-bbox="1145 1100 1300 1158">NULL</td> <td data-bbox="1300 1100 1456 1158">NULL</td> </tr> </tbody> </table>	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</code> ·( <code>[DISTINCT]</code> <code>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</code> ·( <code>expr</code> ) <code>COUNT</code> (*)	Подсчитывает число записей и имеет несколько форм. Форма <code>COUNT</code> ( <code>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</code> ( <code>DISTINCT expr1, expr2, ...</code> ) позволяет подсчитать только уникальные значения полей.
<code>GROUP_CONCAT</code> ·( <code>.</code> )	Функция <code>GROUP_CONCAT</code> ( <code>.</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</code> ·( <code>[DISTINCT]</code> <code>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</code> ·( <code>[DISTINCT]</code> <code>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</code> ·( <code>expr</code> )	Возвращает стандартное среднееквадратичное отклонение в аргументе <code>expr</code> .
<code>STDDEV_SAMP</code> ( <code>expr</code> )	Возвращает выборочное среднееквадратичное отклонение в аргументе <code>expr</code> .
<code>SUM</code> ·( <code>[DISTINCT]</code> <code>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 · (X)	Возвращает абсолютное значение аргумента X
CEIL · (X)	Возвращает наименьшее целочисленное значение, которое больше или равно числу X (CEIL · (X) ≥ X). Пример: CEIL · (12.34) = 13
FLOOR · (X)	Возвращает наибольшее целочисленное значение, которое меньше или равно числу X (FLOOR · (X) ≤ X). Пример: FLOOR · (12.34) = 12
MOD · (M, N)	Возвращает остаток от деления целого числа M на целое число N
POW · (X, Y)	Возвращает значение числа X, возведенного в степень Y (X <sup>Y</sup> = POW(X, Y))
RAND · (X)	Возвращает случайное значение с плавающей точкой (от 0.0 до 1.0)
ROUND · (X, · D)	Возвращает число X, округленное до определенного D количества знаков после запятой (D = 0, 1, 2 . . .)
SIGN · (X)	Определяет знак числа X. Если X отрицательно — возвращает минус один (-1), если положительно — единицу (1), если ноль — возвращает ноль (0)
SQRT · (X)	Вычисляет квадратный корень числа X
TRUNCATE (X, D) D = 0, 1, . . .	Возвращает число X, усеченное до определенного D количества знаков после запятой (D = 0, 1, 2 . . .). 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( <i>str</i> , <i>str_list</i> )	Ищет вхождение строки <i>str</i> в список <i>str_list</i> и возвращает номер строки в этом списке (нумерация начинается с 1). Параметр <i>str_list</i> – набор строк, разделенных запятыми
INSERT( <i>str</i> , <i>pos</i> , <i>len</i> , <i>new_str</i> )	Возвращает строку <i>str</i> , в которой подстрока, начинающаяся с позиции <i>pos</i> и имеющая длину <i>len</i> символов, заменена подстрокой <i>new_str</i>
INSTR( <i>str</i> , <i>substr</i> )	Возвращает позицию первого вхождения подстроки <i>substr</i> в строку <i>str</i>
LEFT( <i>str</i> , <i>len</i> )	Возвращает <i>len</i> крайних левых символов строки <i>str</i>
LENGTH( <i>str</i> )	Возвращает длину строки <i>str</i>
LOCATE( <i>substr</i> , <i>str</i> [, <i>pos</i> ])	Возвращает позицию первого вхождения подстроки <i>substr</i> в строку <i>str</i> . При наличии необязательного аргумента <i>pos</i> , поиск начинается с позиции, указанной в этом аргументе ( <i>pos</i> )
LOWER( <i>str</i> )	Возвращает строку <i>str</i> , 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