

## 2.4. ОПЕРАТОР SELECT

Рассмотрим наиболее популярные запросы к базам данных. Мы не будем углубляться очень детально в специфику SQL-запросов, можно их изучить с помощью учебного пособия [17; 20], документации [2].

Запрос на выборку SELECT позволяет выбрать данные из таблицы с учетом некоторого условия.

Общи вид запроса на выборку следующий:

```
SELECT <что_выбираем>  
FROM <имя_таблицы>  
[WHERE <условие_для_выборки>]  
[ORDER BY <название_поля> [ASC | DESC]]  
[LIMIT [offset,] rows]  
]
```

<что\_выбираем> может быть перечнем полей, либо оператором \*, который заменяет выбор всех полей.

Сравните:

```
SELECT id, title, description, text, author FROM articles
или
SELECT * FROM articles
```

Выражения **WHERE**, **ORDER BY** и **LIMIT** не являются обязательными в разделе. Именно поэтому в нашем запросе они заключены в квадратные скобки.

Выражение **WHERE** позволяет задать условие выборки. Если нужно составить сложное условие, то могут быть использованы операторы **AND** (логическое «И»), **OR** (логическое «ИЛИ») и **NOT** (логическое «НЕ»).

Пример, запрос на выборку всех записей со значением поля id, равное 2, и поля author, содержащего 'Расмус Лерддорф':

```
SELECT * FROM articles WHERE id = 2 AND author='Расмус
Лерддорф'
```

MySQL условие **LIKE** позволяет использовать шаблоны в операторе **WHERE**. Это позволяет выполнять сопоставление шаблонов. В сочетании с оператором **LIKE** используются два подстановочных знака:

- «%» — знак процента представляет ноль, один или несколько символов;
- «\_» — знак подчеркивания представляет один символ.

Примеры работы с операторами представления находятся в таблице 3.

Таблица 3

Примеры операторов представления

LIKE Оператор	Описание
WHERE fieldName LIKE 'a%'	Находит любые значения, которые начинаются с «а»
WHERE fieldName LIKE '%a'	Находит любые значения, которые заканчиваются «а»
WHERE fieldName LIKE '%or%'	Находит любые значения, которые имеют значение «ог» в любом положении
WHERE fieldName LIKE '_r%'	Находит все значения, которые имеют значение «г» во второй позиции
WHERE fieldName LIKE 'a_%_%'	Находит любые значения, которые начинаются с «а» и длиной не менее 3 символов
WHERE fieldName LIKE 'a%o'	Находит любые значения, которые начинаются с «а» и заканчиваются на "о"

Выражение **ORDER BY** используется для сортировки записей в вашем результирующем наборе. Здесь обязательно указывается поле для сортировки и тип сортировки: по возрастанию (**ASC**) и по убыванию (**DESC**). Если атрибуты **ASC** или **DESC** не указаны в операторе **ORDER BY**, результаты будут отсортированы по полю в порядке возрастания. Это эквивалентно выражению **ORDER BY <поле> ASC**.

Например, следующий запрос отсортирует сначала в порядке убывания по автору, затем по названию в порядке возрастания:

```
SELECT * FROM articles ORDER BY author DESC, title
```

Выражение **LIMIT** может использоваться для ограничения количества строк, возвращенных командой **SELECT**. **LIMIT** принимает один или два числовых аргумента. Эти аргументы должны быть целочисленными константами. Если заданы два аргумента, то первый указывает на начало первой возвращаемой строки, а второй задает максимальное количество возвращаемых строк. При этом смещение начальной строки равно 0 (не 1).

```
SELECT * FROM articles LIMIT 5,10
```

возвращает строки 6–15.

Часто возникает необходимость выбрать из базы данных некоторые строки в случайном порядке и вывести только часть из них. Например, на сайте в фотогалерее требуется вывести 5 случайных фотографий. Для этого MySQL имеет встроенную функцию **RAND()**. Работает она следующим образом:

```
SELECT * FROM articles ORDER BY RAND() LIMIT 5
```

Здесь функция **RAND()** генерирует случайное число для каждой строки в таблице. Предложение **ORDER BY** сортирует все строки в таблице по случайному числу, сгенерированному функцией **RAND()**. Предложение **LIMIT** выбирает первые 5 строк в наборе результатов, отсортированных случайным образом. Если **LIMIT** не указан, то будут выведены все записи в случайном порядке. Данный способ окажется гораздо быстрее по скорости и лаконичнее по записи кода, чем если бы мы доверили ту же задачу PHP.

Если требуется отсортировать записи в некотором заранее известном порядке для разработчика (сортировка по определенной последовательности), то применяется функция FIELD(). В данной функции на первом месте указывается поле, по которому производится сортировка, затем через запятую перечисляется порядок, в котором должны выводиться записи по указанному полю.

```
SELECT * FROM articles ORDER BY FIELD(id, 4,2,1,3)
```

Данный запрос выведет первые записи из таблицы **articles** в порядке поля **id – 4, 2, 1, 3** записи. Если в таблице есть другие записи, то заданная упорядоченность работать не будет. Если же значений в функции FIELD() больше, чем строк в таблице, то заданный порядок сохранится.

К запросу на выборку применимы также и некоторые агрегатные функции. К ним относят подсчет суммы, количества, минимального значения, максимального значения, среднего арифметического. Агрегатные функции SQL действуют в отношении значений столбца с целью получения единого результирующего значения. Проще говоря, используя эти функции, мы можем из множества значений столбца получить одно единственное. Такие же операции программист может сделать и с использованием языка программирования, однако MySQL выполняет подобные запросы на порядок быстрее.

Общий формат обращения к агрегатной функции в операторе SELECT следующий:

```
SELECT <функция>(<поле>) FROM <таблица> [WHERE ...]
```

В качестве функций может выступать одно из следующих значений

- AVG – вычисляет среднее значение;
- SUM – вычисляет сумму значений;
- MIN – вычисляет наименьшее значение;
- MAX – вычисляет наибольшее значение;
- COUNT – вычисляет количество строк в запросе.

Пусть имеется таблица со списком статей **articles**. Чтобы подсчитать общее количество статей в таблице, можно написать запрос:

```
SELECT COUNT(id) FROM articles
```

В данном случае запрос вернет нам количество записей из таблицы **articles**. Можно записать этот запрос несколько иначе – вместо поля **id** использовать символ \* (все поля). Результат получится одинаковым.