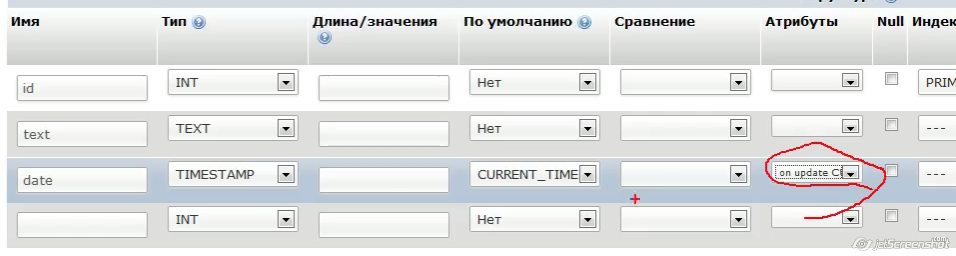
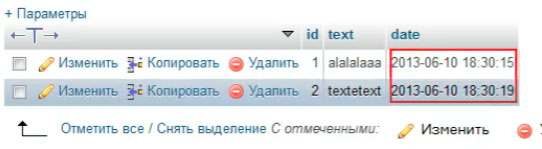
**УРОК 29**

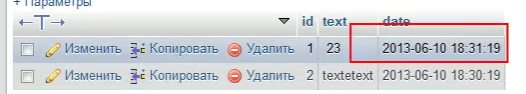
Открываем БД и устанавливаем в поле date значения, для автоматической простановки даты. Дата устанавливается по умолчанию, так которая установлена в БД



Вот как это будет выглядеть после создания записи

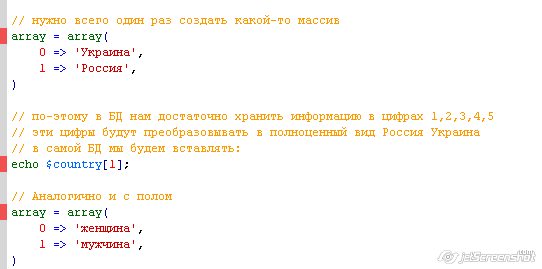


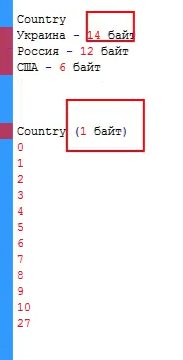
Если мы обновим запись, то автоматически и обновится дата

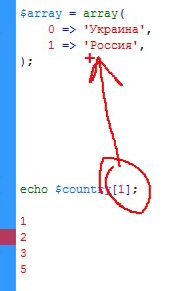


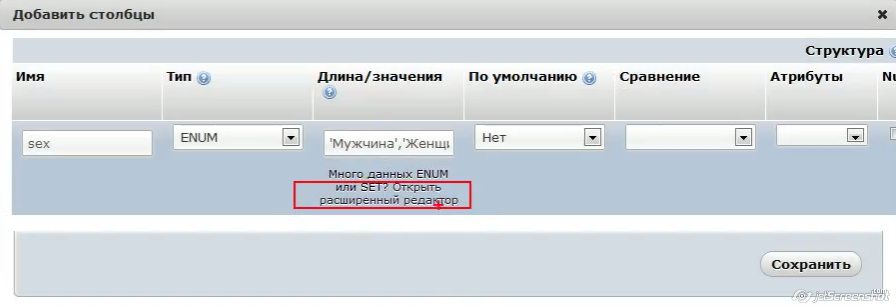
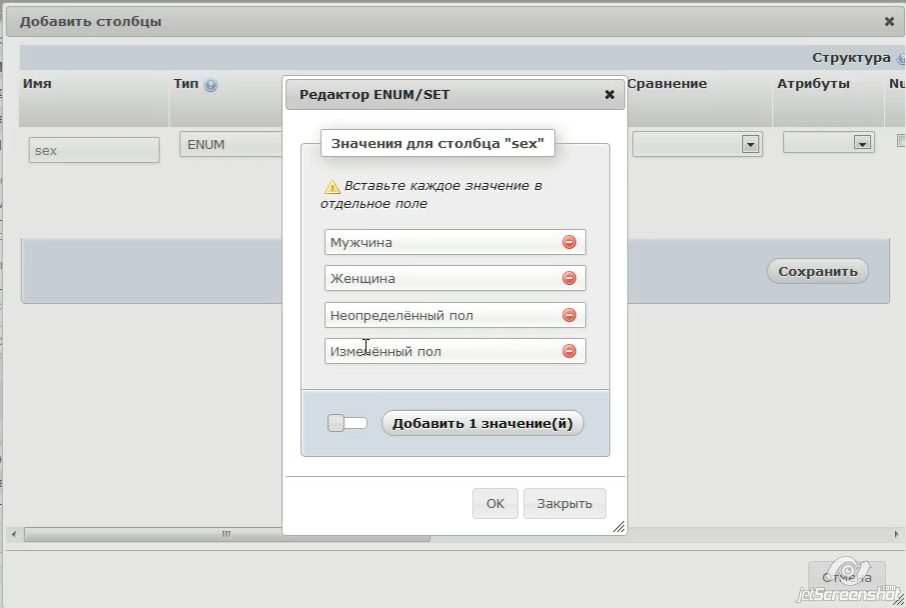
Но мы можем и сами вручную пописывать дату, автоматически – это для удобства.

D:\server\home\next\www\work\lessons_word\29\2014-11-24_11-31_docx.jpg

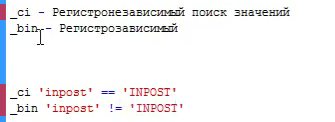


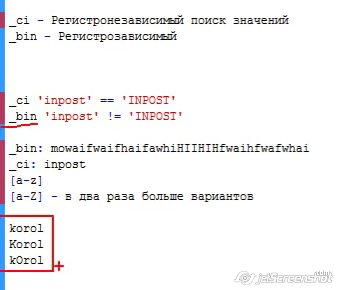


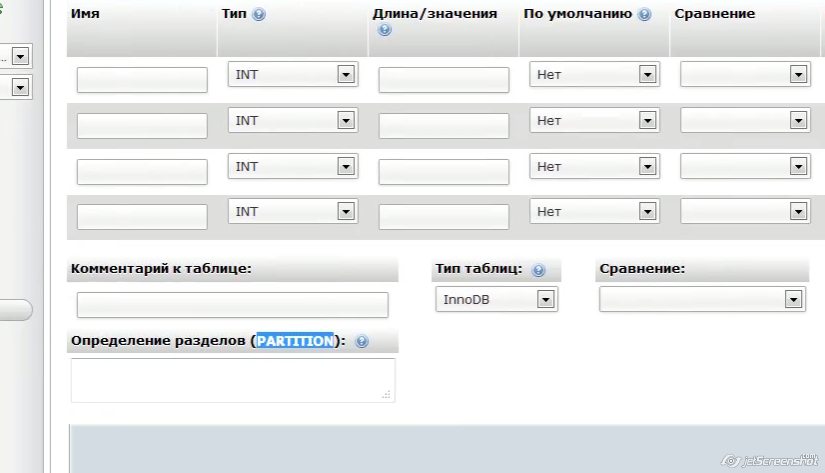
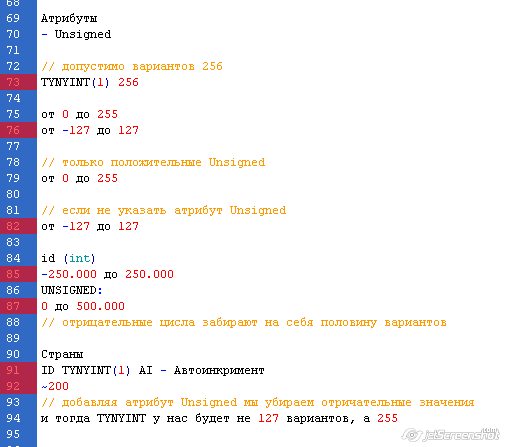


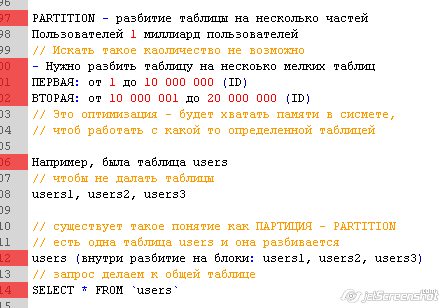


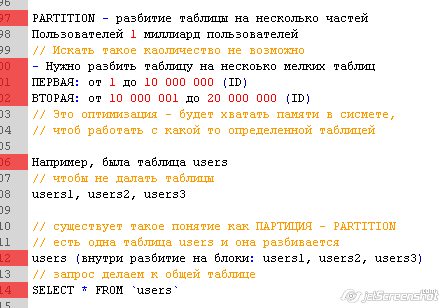
**кодировка**

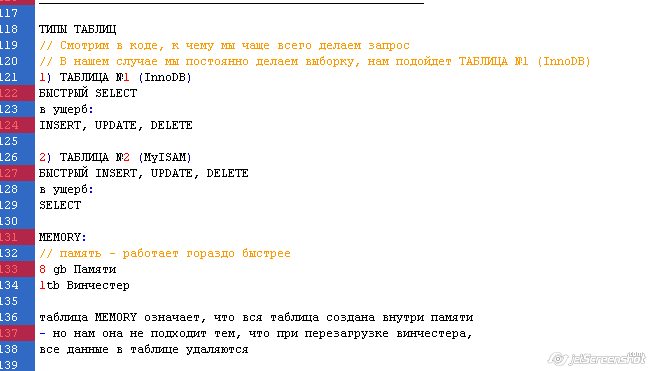




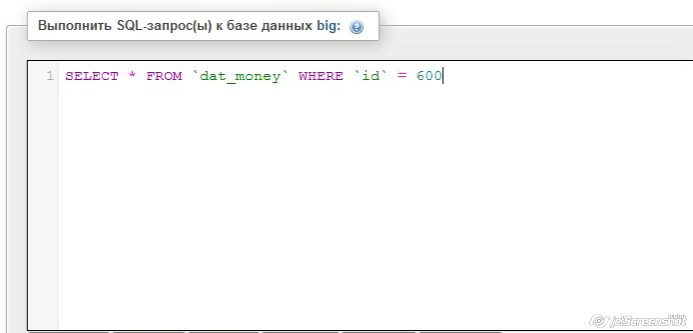


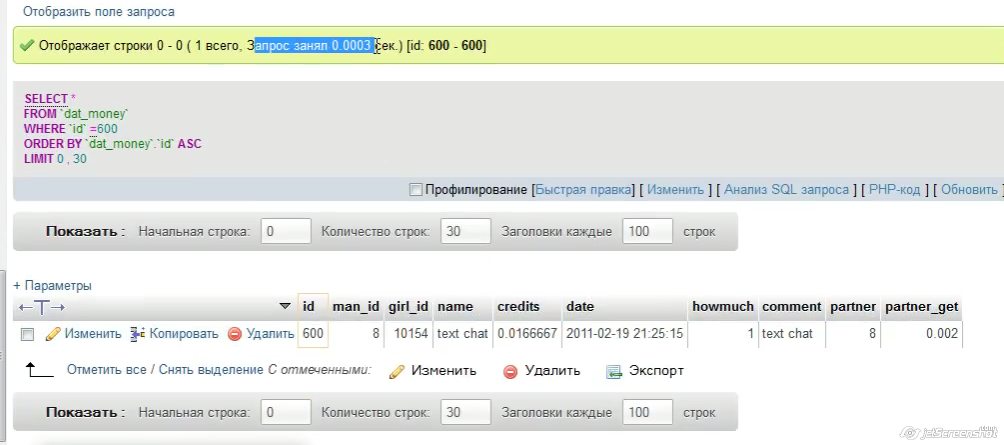




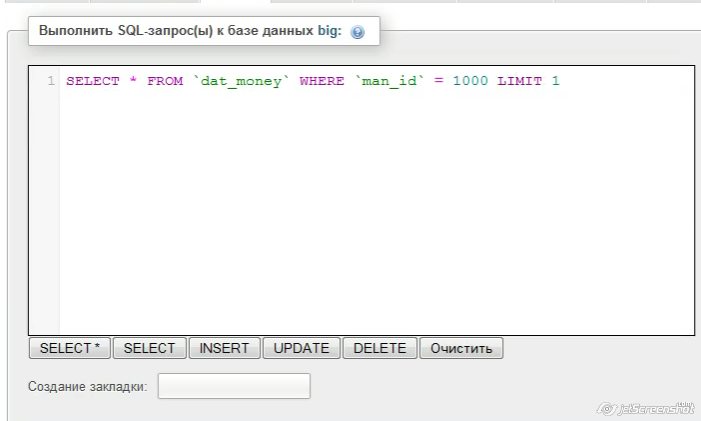


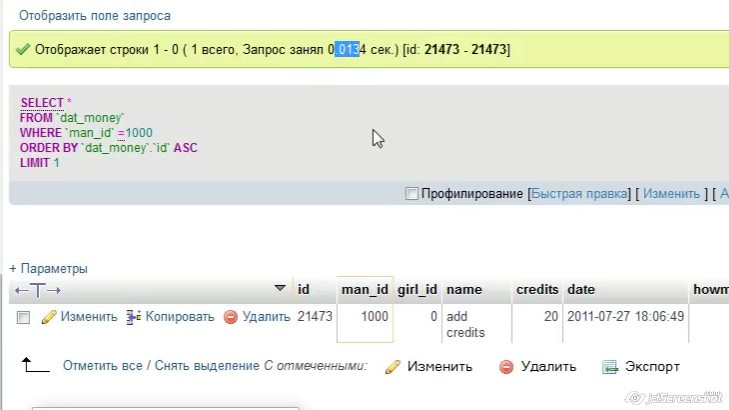
Ищем там, где установлен индекс



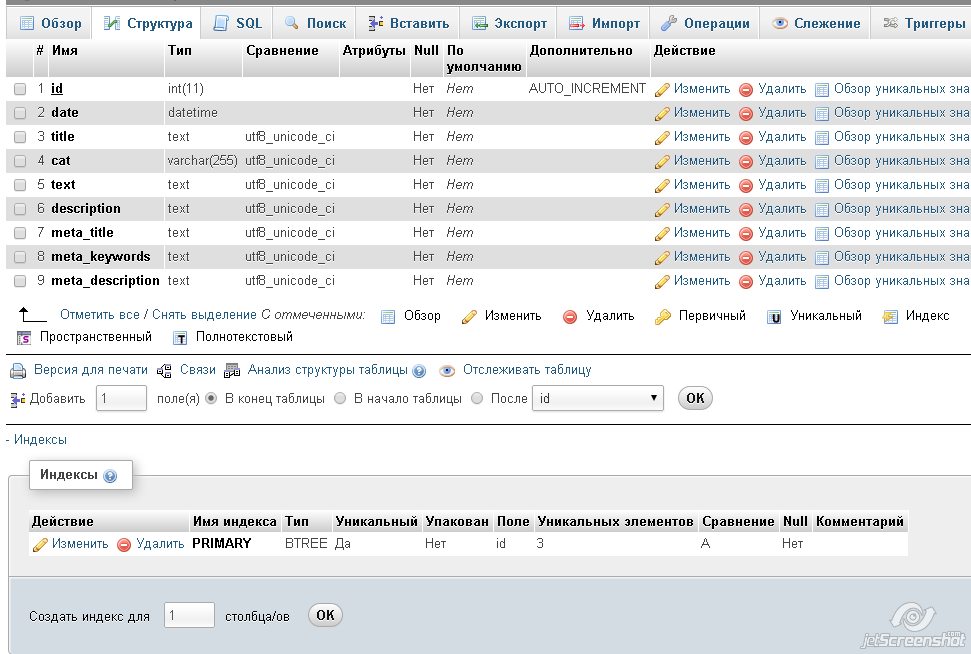


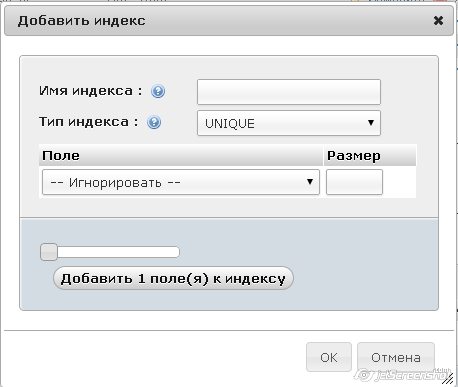
Ищем там, где нет индекса

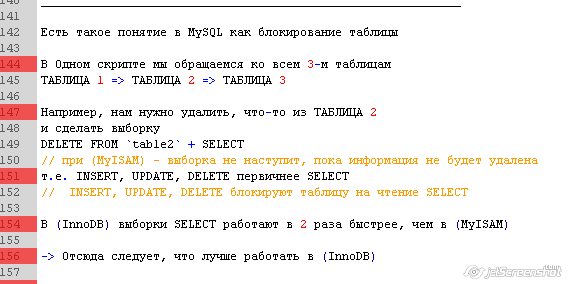




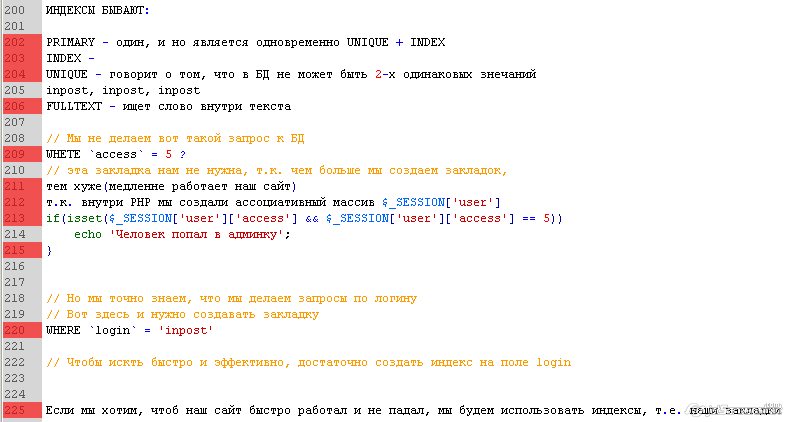
Добавляем индекс





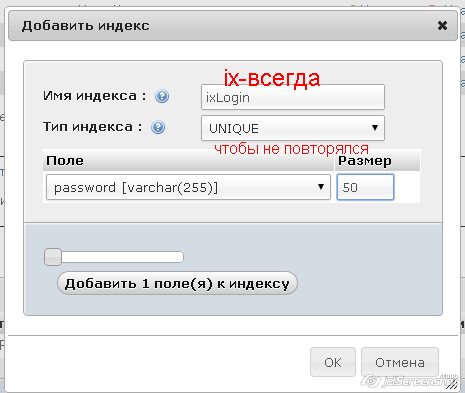






При попытке добавить логин который уже существует будет выводиться ошибка,

т.к. мы установили тип индекса: UNIQUE



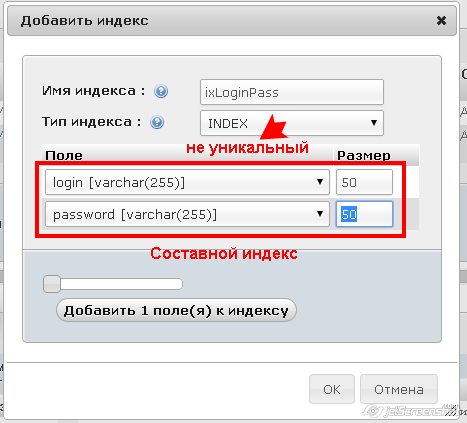
Таким образом, 2-х одинаковых логинов в нашей БД никогда не будет

// Для каждого запроса WHERE устанавливаем индекс

1. WHERE `login` = 'inpost' // уже создали

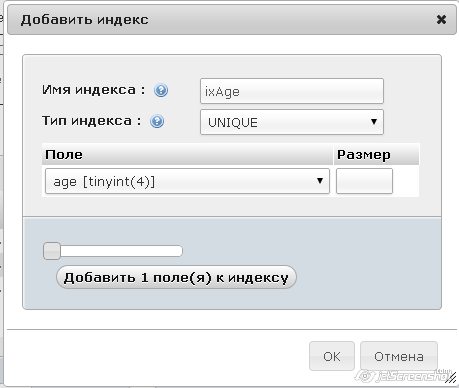
// создаем индексы для 2-х столбцов

1. WHERE `login` = 'inpost' AND `password` = 'lalala'

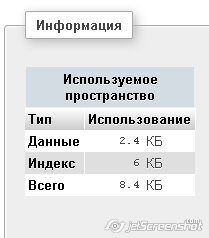


3. WHERE `age` BETWEEN 18 AND 25

// BETWEEN тоже что age >= 18 AND age <= 25

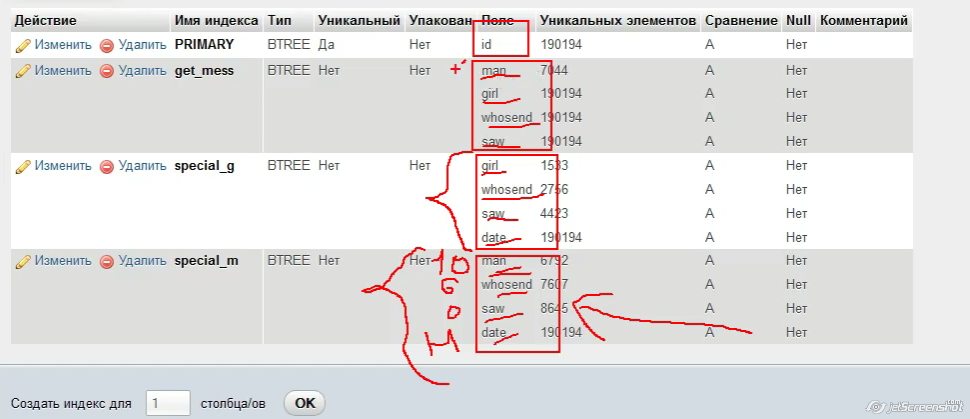


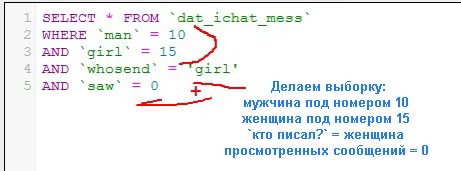
// Несколько индексов занимают памяти больше, чем сама структура БД



// Но теперь у нас условия выборки только по установленным полям

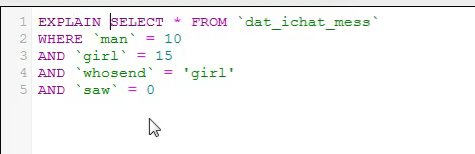




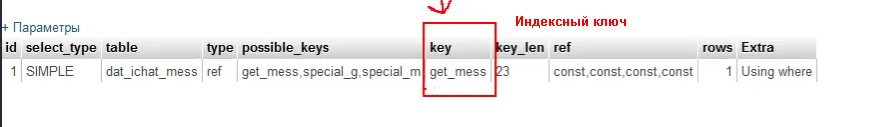


// Чтобы проверить будут в данном случае использоваться индексы или не будут, мы можем

Проанализировать данный запрос при помощи **EXPLANE**

****

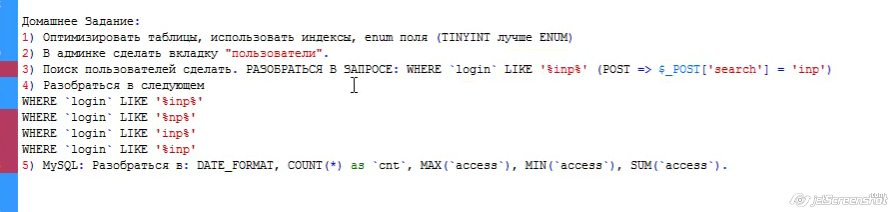
// И мы видим какой ключ используется:

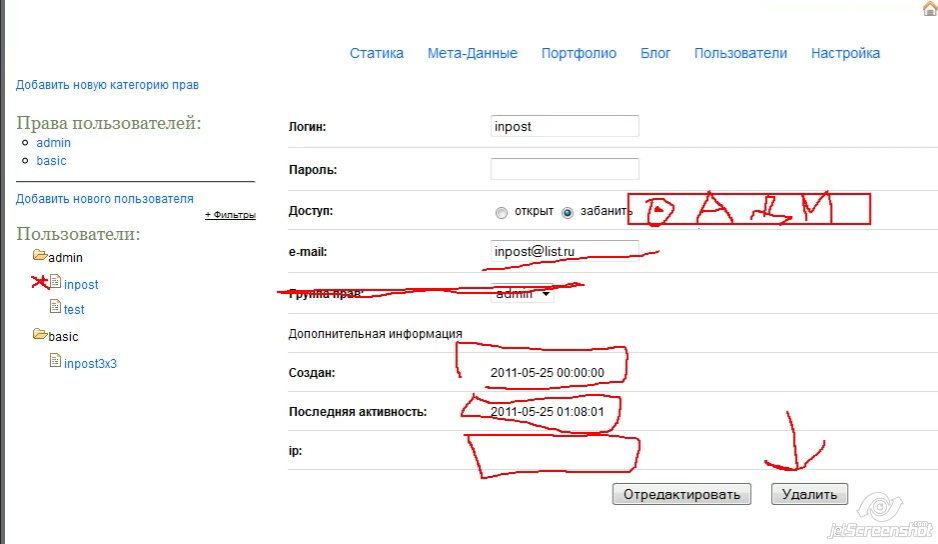


// Ключ не используется **NULL ->** в связи с этим запрос будет очень, очень медленный



при помощи **EXPLANE** мы анализирует наши запросы и видим, какие из них оптимизированы, а какие нет!





4. Разобраться в следующем:  
WHERE `login` LIKE '%inp%'

 Помимо знаков равенства,  также "больше" или "меньше", в языке MySQL запросов существует понятие "**похоже на**". Выберем записи из таблицы my\_sql\_table, где в фамилии встречается "*дор*" :

$query = "SELECT \* FROM `my\_sql\_table` where surname like '%дор%'";

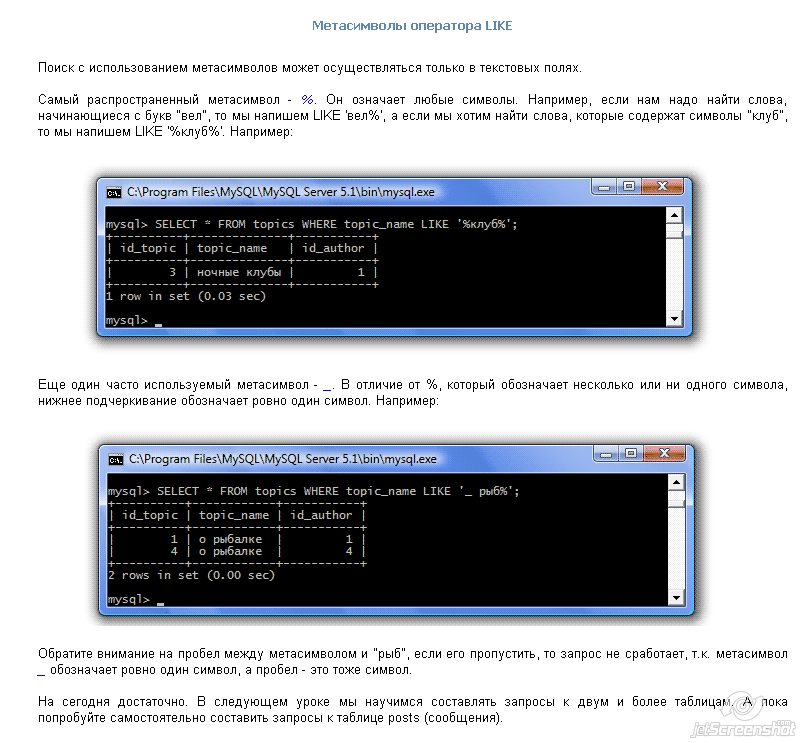
Здесь наличие "*%*" в начале и конце 'дор' и означает, что запрос будет искать именно 'дор', причем не важно в начале, конце, или середине фамилии он находится. Рассмотрим следующий пример

WHERE `login` LIKE 'inp%'

5. Выберем записи из таблицы my\_sql\_table с фамилией, которая начинается на **П**. Обратите внимание на расположение "%":

$query = "SELECT \* FROM `my\_sql\_table` where surname like 'П%'";

WHERE `login` LIKE '%inp'



SQL Like

[PDF](http://dimonchik.com/sql-like.pdf)[E-mail](http://dimonchik.com/component/mailto/?tmpl=component&link=84a516d444d3da82c2649f35e199ee6664af7908)[Печать](http://dimonchik.com/sql-like.html?tmpl=component&print=1&page=)14.02.2009

**LIKE**- данный оператор используется с условием WHERE для поиска значений по образцу.

**Синтаксис SQL LIKE**

|  |  |
| --- | --- |
| 1  2  3 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) column\_name(s)  [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name  [**WHERE**](http://search.mysql.com/search?site=refman-51&q=WHERE&) column\_name [**LIKE**](http://dev.mysql.com/doc/refman/5.1/en/non-typed-operators.html) pattern |

**Пример 1**

Есть таблица "Persons":

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_Id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |
| 3 | Pettersen | Kari | Storgt 20 | Stavanger |

Теперь мы хотим выбрать всех лиц, которые живут в городе, имя которого**начинается** с буквы "s" из таблицы выше.

Для этого используем такой запрос:

|  |  |
| --- | --- |
| 1  2 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) \* [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Persons  [**WHERE**](http://search.mysql.com/search?site=refman-51&q=WHERE&) City [**LIKE**](http://dev.mysql.com/doc/refman/5.1/en/non-typed-operators.html) 's**%**' |

**"%"** - данный знак является маской. Обозначает любые символы.

Результат запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_Id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |
| 3 | Pettersen | Kari | Storgt 20 | Stavanger |

**Пример 2**

Теперь мы хотим выбрать всех лиц, которые живут в городе, имя которого **заканчивается** на букву "s" из таблицы выше.

Для этого используем такой запрос:

|  |  |
| --- | --- |
| 1  2 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) \* [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Persons  [**WHERE**](http://search.mysql.com/search?site=refman-51&q=WHERE&) City [**LIKE**](http://dev.mysql.com/doc/refman/5.1/en/non-typed-operators.html) '**%**s' |

Результат запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_Id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |

**Пример 3**

Теперь мы хотим выбрать всех лиц, которые живут в городе, имя которого **содержит** в себе буквы "tav". Не важно, в каком месте будут находиться данные символы, в начале или конце слова.

Для этого используем такой запрос.:

|  |  |
| --- | --- |
| 1  2 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) \* [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Persons  [**WHERE**](http://search.mysql.com/search?site=refman-51&q=WHERE&) City [**LIKE**](http://dev.mysql.com/doc/refman/5.1/en/non-typed-operators.html) '**%**tav**%**' |

Результат запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_Id** | **LastName** | **FirstName** | **Address** | **City** |
| 3 | Pettersen | Kari | Storgt 20 | Stavanger |

**Пример 4**

Теперь мы хотим выбрать всех лиц, которые живут в городе, имя которого**не** содержит в себе буквы "tav". Используем ключевое слово NOT.

Для этого используем такой запрос.:

|  |  |
| --- | --- |
| 1  2 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) \* [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Persons  [**WHERE**](http://search.mysql.com/search?site=refman-51&q=WHERE&) City [**NOT**](http://dev.mysql.com/doc/refman/5.1/en/non-typed-operators.html) [**LIKE**](http://dev.mysql.com/doc/refman/5.1/en/non-typed-operators.html) '**%**tav**%**' |

Результат запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_Id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |

Также можно использовать символ подчеркивания (\_) для замены отдельного символа в значении. Например, если вы зададите:   
  
SELECT \*   
FROM Persons  
WHERE FirstName LIKE 'To\_'  
  
То результатом запроса будет значение Tom, Tos, Top но не Tove, поскольку в нем четыре символа, а вы задали в поиск значение с тремя.

5. MySQL: Разобраться в:

DATE\_FORMAT

Формат использования: DATE\_FORMAT(date,'format')

где date- поле БД,  'format' - формат вывода.

как написать формат:

|  |  |  |
| --- | --- | --- |
| %M | Month name (January..December) | Имя месяца на английском языке |
| %W | Weekday name (Sunday..Saturday) | День недели на английском |
| %D | Day of the month with English suffix (1st, 2nd, 3rd, etc.) | День месяца на английском |
| %Y | Year, numeric, 4 digits | Год, 1991, 2006 |
| %y | Year, numeric, 2 digits | Год, 91, 06 |
| %X | Year for the week where Sunday is the first day of the week, numeric, 4 digits, used with '%V' | Год для тех, у кого неделя начинается в воскресенье |
| %x | Year for the week, where Monday is the first day of the week, numeric, 4 digits, used with '%v' | Год для тех, у кого неделя начинается в понедельник |
| %a | Abbreviated weekday name (Sun..Sat) | Сокращенный день недели на англ. (Sun..Sat) |
| %d | Day of the month, numeric (00..31) | День месяца, с нулем вначале |
| %e | Day of the month, numeric (0..31) | День месяца, без нуля вначале |
| %m | Month, numeric (01..12) | Месяц,  с нулем вначале |
| %c | Month, numeric (1..12) | Месяц, без нуля вначале |
| %b | Abbreviated month name (Jan..Dec) | Месяц на английском |
| %j | Day of year (001..366) | День года (001..366) |
| %H | Hour (00..23) | Час, с нулем вначале |
| %k | Hour (0..23) | Час, без нуля вначале |
| %h | Hour (01..12) | Час в 12-часовой системе, с нулем вначале |
| %I | Hour (01..12) | Час в 12-часовой системе, с нулем вначале |
| %l | Hour (1..12) | Час в 12-часовой системе, без нуля вначале |
| %i | Minutes, numeric (00..59) | Минуты |
| %r | Time, 12-hour (hh:mm:ss [AP]M) | Время в 12-часовой системе |
| %T | Time, 24-hour (hh:mm:ss) | Время в 24-часовой системе |
| %S | Seconds (00..59) | Секунды |
| %s | Seconds (00..59) | Секунды |
| %p | AM or PM | Ночь или день, в английской системе |
| %w | Day of the week (0=Sunday..6=Saturday) | День недели, цифра |
| %U | Week (0..53), where Sunday is the first day of the week | Номер недели с начала года, неделя начинается в воскресенье |
| %u | Week (0..53), where Monday is the first day of the week | Номер недели с начала года, неделя начинается в понедельник |
| %V | Week (1..53), where Sunday is the first day of the week. Used with '%X' | Номер недели |
| %v | Week (1..53), where Monday is the first day of the week. Used with '%x' | Номер недели |
| %% | A literal `%'. |  |

Пример:

**SELECT \*,DATE\_FORMAT(date\_registration,'%d.%m.%Y')  as date\_reg  FROM users**

даты будут в следующем формате: 04.09.2007  и будут находиться в поле **date\_reg**, не**date\_registration** (в возвращаемом массиве будет новое поле **date\_reg**)

# Преобразование даты в mysql с помощью DATE\_FORMAT()

Как известно, все даты хранятся в mysql в обратном порядке год-месяц-число(2008-10-18), иногда даже без разделителя(20081018).  
Чтобы вывести дату, нужно ее преобразовать в нормальный читаемый вид.

Тут есть два способа преобразования, эффективный и не очень.  
Не эффктивный способ это когда дату выводимую из mysql преобразуют с помощью php.  
Сам лично так делал очень долгое время. Перед выводом на экран переворачивал каждую дату с помощью php функции.   
Если количество преобразований не большое, то можно переворачивать дату и с помощью php, ничего плохого тут нет, но если нужно будет выдернуть десятки-сотни тысяч записей и в каждой преобразовать дату, то тут конечно намного быстрее будет преобразование дат с помощью mysql.

В mysql существует отличная функция DATE\_FORMAT(), она очень похожа на php функцию date().  
Вот пример использования   
  
SELECT DATE\_FORMAT("2008-11-19",'%d.%m.%Y');  
результат  
  
19.11.2008

Все очень просто и быстро, нет необходимости крутить даты с помощью php.  
Вот список определителей этой функции

|  |  |
| --- | --- |
| **Определитель** | **Описание** |
| %M | Название месяца (январь...декабрь) |
| %W | Название дня недели (воскресенье...суббота) |
| %D | День месяца с английским суффиксом (0st, 1st, 2nd, 3rd и т.д.) |
| %Y | Год, число, 4 разряда |
| %y | Год, число, 2 разряда |
| %X | Год для недели, где воскресенье считается первым днем недели, число, 4 разряда, используется с '%V' |
| %x | Год для недели, где воскресенье считается первым днем недели, число, 4 разряда, используется с '%v' |
| %a | Сокращенное наименование дня недели (Вс...Сб) |
| %d | День месяца, число (00..31) |
| %e | День месяца, число (0..31) |
| %m | Месяц, число (00..12) |
| %c | Месяц, число (0..12) |
| %b | Сокращенное наименование месяца (Янв...Дек) |
| %j | День года (001..366) |
| %H | Час (00..23) |
| %k | Час (0..23) |
| %h | Час (01..12) |
| %I | Час (01..12) |
| %l | Час (1..12) |
| %i | Минуты, число (00..59) |
| %r | Время, 12-часовой формат (hh:mm:ss [AP]M) |
| %T | Время, 24-часовой формат (hh:mm:ss) |
| %S | Секунды (00..59) |
| %s | Секунды (00..59) |
| %p | AM или PM |
| %w | День недели (0=воскресенье..6=суббота) |
| %U | Неделя (00..53), где воскресенье считается первым днем недели |
| %u | Неделя (00..53), где понедельник считается первым днем недели |
| %V | Неделя (01..53), где воскресенье считается первым днем недели. Используется с `%X' |
| %v | Неделя (01..53), где понедельник считается первым днем недели. Используется с `%x' |
| %% | Литерал `%'. |

COUNT(\*) as `cnt`

SQL count()

[PDF](http://dimonchik.com/sql-count.pdf)[E-mail](http://dimonchik.com/component/mailto/?tmpl=component&link=bd2d8ea35059fcdb03c99176b3598b4ae40a2d34)[Печать](http://dimonchik.com/sql-count.html?tmpl=component&print=1&page=)14.02.2009

**COUNT()** - данная функция возвращает количество строк, которые соответствует определенным критериям.

**Синтаксис SQL COUNT(column\_name)**

COUNT(COLUMN\_NAME) - функция возвращает определенная количество записей (NULL значения не будут учитываться) конкретного столбца:

[**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [COUNT](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(column\_name) [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name

**Синтаксис SQL COUNT(\*)**

В COUNT (\*) - функция возвращает все записи в таблице:

[**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [COUNT](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(\*) [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name

**Синтаксис SQL COUNT(DISTINCT column\_name)**

COUNT(DISTINCT COLUMN\_NAME) - возвращает только разные записи:

[**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [COUNT](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)([**DISTINCT**](http://search.mysql.com/search?site=refman-51&q=DISTINCT&) column\_name) [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name

Note - **Замечание:** COUNT (DISTINCT) работает в ORACLE и Microsoft SQL Server, но не в Microsoft Access.

**Пример SQL COUNT(column\_name)**

Есть следующая таблица "Orders":

|  |  |  |  |
| --- | --- | --- | --- |
| **O\_Id** | **OrderDate** | **OrderPrice** | **Customer** |
| 1 | 2008/11/12 | 1000 | Hansen |
| 2 | 2008/10/23 | 1600 | Nilsen |
| 3 | 2008/09/02 | 700 | Hansen |
| 4 | 2008/09/03 | 300 | Hansen |
| 5 | 2008/08/30 | 2000 | Jensen |
| 6 | 2008/10/04 | 100 | Nilsen |

Теперь мы хотим подсчитать число заказов от клиента "Nilsen".

Выполним запрос:

|  |  |
| --- | --- |
| 1  2 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [COUNT](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(Customer) [**AS**](http://search.mysql.com/search?site=refman-51&q=AS&) CustomerNilsen [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Orders  [**WHERE**](http://search.mysql.com/search?site=refman-51&q=WHERE&) Customer='Nilsen' |

Результат запроса:

|  |
| --- |
| **CustomerNilsen** |
| 2 |

**Пример SQL COUNT(\*)**

Если не прописать условие WHERE Е.

|  |  |
| --- | --- |
| 1 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [COUNT](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(\*) [**AS**](http://search.mysql.com/search?site=refman-51&q=AS&) NumberOfOrders [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Orders |

Результат будет такой:

|  |
| --- |
| **NumberOfOrders** |
| 6 |

**Пример SQL COUNT(DISTINCT column\_name)**

Теперь мы хотим подсчитать количество уникальных клиентов в таблице "Orders".

Выполним запрос:

|  |  |
| --- | --- |
| 1 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [COUNT](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)([**DISTINCT**](http://search.mysql.com/search?site=refman-51&q=DISTINCT&) Customer) [**AS**](http://search.mysql.com/search?site=refman-51&q=AS&) NumberOfCustomers [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Orders |

Результат запроса:

|  |
| --- |
| **NumberOfCustomers** |
| 3 |

7. Вычислим количество полей в my\_sql\_table с фамилией, которая начинается на **П**.

$query = "SELECT COUNT(\*) FROM `my\_sql\_table` where surname like 'П%'";

MAX(`access`)

6. Вычислим максимальное значение **id**:

$query = "SELECT MAX(id) FROM `my\_sql\_table`";

SQL max()

[PDF](http://www.dimonchik.com/sql-max.pdf)[E-mail](http://www.dimonchik.com/component/mailto/?tmpl=component&link=b758bfc4987860fef6b4bd9c6ee0cef210b1e13b)[Печать](http://www.dimonchik.com/sql-max.html?tmpl=component&print=1&page=)14.02.2009

**MAX()** - функция возвращает наибольшее значение для выбранного столбца.

**Синтаксис SQL MAX()**

[**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [MAX](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(column\_name) [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name

**Пример SQL MAX()**

Есть следующая таблица "Orders":

|  |  |  |  |
| --- | --- | --- | --- |
| **O\_Id** | **OrderDate** | **OrderPrice** | **Customer** |
| 1 | 2008/11/12 | 1000 | Hansen |
| 2 | 2008/10/23 | 1600 | Nilsen |
| 3 | 2008/09/02 | 700 | Hansen |
| 4 | 2008/09/03 | 300 | Hansen |
| 5 | 2008/08/30 | 2000 | Jensen |
| 6 | 2008/10/04 | 100 | Nilsen |

Теперь мы хотим найти наибольшее значение колонки "OrderPrice".

Выполним запрос:

|  |  |
| --- | --- |
| 1 | SELECT [MAX](http://www.php.net/max%3E%3Cspan%20style=)(OrderPrice) AS LargestOrderPrice FROM Orders |

Результат запроса:

|  |
| --- |
| **LargestOrderPrice** |
| 2000 |

MIN(`access`)

SQL min()

[PDF](http://dimonchik.com/sql-min.pdf)[E-mail](http://dimonchik.com/component/mailto/?tmpl=component&link=01ae2ee8e3bd0c92dc67d193c7a845087fe25be2)[Печать](http://dimonchik.com/sql-min.html?tmpl=component&print=1&page=)14.02.2009

**MIN()** - функция возвращает наименьшее значение для выбранного столбца.

**Синтаксис SQL MIN()**

[**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [MIN](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(column\_name) [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name

**Пример SQL MIN()**

Есть следующая таблица "Orders":

|  |  |  |  |
| --- | --- | --- | --- |
| **O\_Id** | **OrderDate** | **OrderPrice** | **Customer** |
| 1 | 2008/11/12 | 1000 | Hansen |
| 2 | 2008/10/23 | 1600 | Nilsen |
| 3 | 2008/09/02 | 700 | Hansen |
| 4 | 2008/09/03 | 300 | Hansen |
| 5 | 2008/08/30 | 2000 | Jensen |
| 6 | 2008/10/04 | 100 | Nilsen |

Теперь мы хотим найти наименьшее значение из колонки "OrderPrice".

Выполним запрос:

|  |  |
| --- | --- |
| 1 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [MIN](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(OrderPrice) [**AS**](http://search.mysql.com/search?site=refman-51&q=AS&) SmallestOrderPrice [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Orders |

Результат запроса:

|  |
| --- |
| **SmallestOrderPrice** |
| 100 |

SUM(`access`)

SQL sum()

[PDF](http://dimonchik.com/sql-sum.pdf)[E-mail](http://dimonchik.com/component/mailto/?tmpl=component&link=f800dde15a32ddc6ad175a0ee7b9c70a9d9e64aa)[Печать](http://dimonchik.com/sql-sum.html?tmpl=component&print=1&page=)14.02.2009

**SUM()**- функция возвращает сумму числового столбца.

**Синтаксис SQL SUM()**

[**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [SUM](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(column\_name) [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) table\_name

**Пример SQL SUM()**

Есть следующая таблица "Orders":

|  |  |  |  |
| --- | --- | --- | --- |
| **O\_Id** | **OrderDate** | **OrderPrice** | **Customer** |
| 1 | 2008/11/12 | 1000 | Hansen |
| 2 | 2008/10/23 | 1600 | Nilsen |
| 3 | 2008/09/02 | 700 | Hansen |
| 4 | 2008/09/03 | 300 | Hansen |
| 5 | 2008/08/30 | 2000 | Jensen |
| 6 | 2008/10/04 | 100 | Nilsen |

Теперь мы хотим найти сумму всех записей"OrderPrice".

Выполним запрос:

|  |  |
| --- | --- |
| 1 | [**SELECT**](http://search.mysql.com/search?site=refman-51&q=SELECT&) [SUM](http://dev.mysql.com/doc/refman/5.1/en/group-by-functions-and-modifiers.html)(OrderPrice) [**AS**](http://search.mysql.com/search?site=refman-51&q=AS&) OrderTotal [**FROM**](http://search.mysql.com/search?site=refman-51&q=FROM&) Orders |

Результат запроса:

|  |
| --- |
| **OrderTotal** |
| 5700 |

