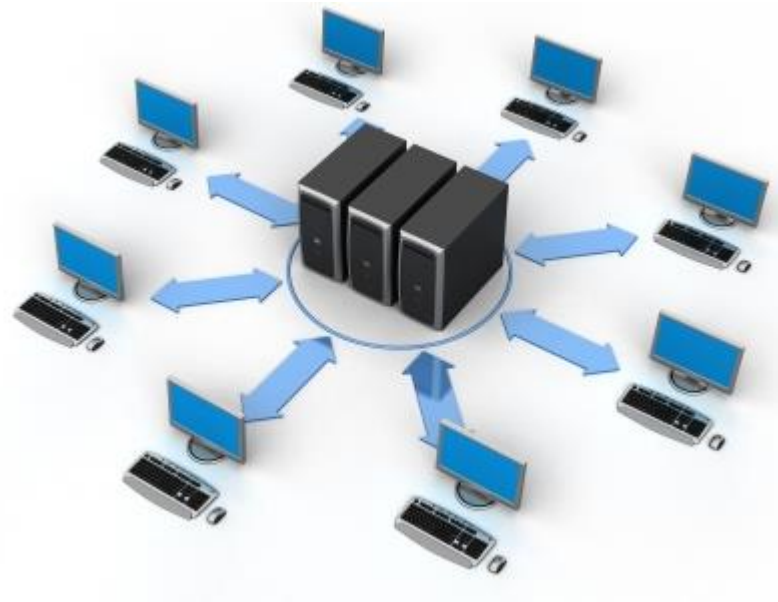


Технологии распределенных систем и параллельных вычислений

Лекция 1: Основы MySQL



Евгений Беркунский,
кафедра ИУСТ, НУК
<http://www.berkut.mk.ua>
eugeny.berkunsky@gmail.com

Програма

- Введение в MySQL
- Простой выбор - Select
- Подключение к MySQL
- Комментарии
- Чувствительность к регистру
- Выбор всего и вся (SELECT All Columns in All Rows)
- Выбор заданных колонок
- Сортировка записей



Програма

- Раздел WHERE и символы операций
- Проверка на равенство и неравенство
- Проверка «на больше-меньше»
- Проверка на NULL
- Раздел WHERE и слова-операторы
- Оператор BETWEEN
- Оператор IN
- Оператор LIKE
- Оператор NOT
- Проверка нескольких условий
- AND, OR, порядок вычислений

Введение в MySQL

- MySQL - быстрая, легкая в использовании СУБД предназначена для применения для множества малых и больших предприятий. MySQL была создана и спонсируется одной фирмой, шведской компанией **MySQL AB**, в настоящее время принадлежит **Oracle Corporation**.
- MySQL стала столь популярной ввиду множества причин.
- MySQL выпускается под open-source лицензией. Поэтому, за ее использование не нужно платить.
- MySQL – мощная программа в своей области. Она поддерживает большое подмножество функциональности более дорогих и мощных СУБД.
- MySQL использует стандартную форму языка SQL.
- MySQL работает на многих ОС и с многими языками программирования, включая PHP, PERL, C, C++, JAVA и т.д.

Введение в MySQL

- MySQL работает очень быстро и хорошо, даже с большими наборами данных.
- MySQL поддерживает большие БД, до **50 млн строк** или даже большее, в таблице.
- По умолчанию, размер таблицы ограничен **4GB**, но вы можете увеличить его (если ваша ОС может обрабатывать такие файлы) до теоретического предела **8 эксабайт (8 млн TB)**.
- **MySQL настраиваемая.** Open-source GPL лицензия разрешает программистам изменять MySQL для подстройки к своему специфичному окружению.

MariaDB

- **MariaDB** — ответвление СУБД MySQL, разрабатываемое сообществом.
- Толчком к созданию стала необходимость обеспечения свободного статуса СУБД (под лицензией GPL), в противовес неопределенной политике лицензирования MySQL компанией Oracle
- Ведущий разработчик — Майкл Видениус, автор оригинальной версии MySQL и основатель компании Monty Program AB



Простой выбор: SELECT

- Запрос на выборку данных

```
SELECT last_name FROM person
```

Подключение к MySQL

- В Java используется статический метод класса DriverManager:

```
Connection conn =  
DriverManager.getConnection(<url>, <properties>);
```


Подключение к MySQL

```
String url = "jdbc:mysql://localhost/example";  
String user = ...;  
String password = ...;  
Properties props = new Properties();  
props.setProperty("user", user);  
props.setProperty("password", password);  
props.setProperty("useUnicode", "true");  
props.setProperty("characterEncoding", "utf8");  
Connection conn = DriverManager.getConnection(url, props);
```

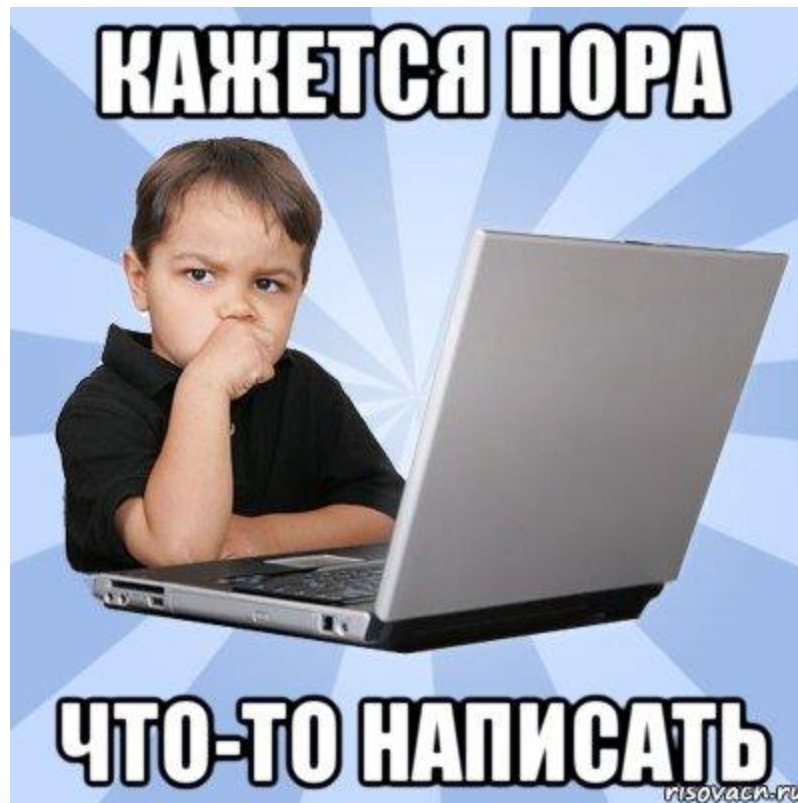
Комментарии

```
# this is a comment  
-- This is also a comment  
/*  
This  
is a  
comment  
*/  
SELECT * FROM book;
```



НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрація



Чувствительность к регистру

- Хотя имена **баз данных, таблиц и триггеров** нечувствительны к регистру на некоторых платформах, не следует на это полагаться и использовать разные регистры в одном операторе.

```
SELECT * FROM person WHERE PERSON.id=1;
```

Чувствительность к регистру

- **Имена столбцов, индексов, хранимых процедур и событий** нечувствительны к регистру на всех платформах.
- По умолчанию, псевдонимы таблиц чувствительны к регистру на Unix, но не на Windows или Mac OS X. Следующий оператор не будет работать на Unix, потому, что он использует псевдоним «a» и «A»:

```
SELECT col_name FROM tbl_name AS a WHERE  
a.col_name = 1 OR A.col_name = 2;
```

- Синтаксис SELECT

```
SELECT expressions_and_columns FROM table_name  
[WHERE some_condition_is_true]  
[ORDER BY some_column [ASC | DESC]]  
[LIMIT offset, rows]
```

```
SELECT * FROM tbl_name;
```

Выбор указанных столбцов

- Если вы не хотите увидеть все столбцы вашей таблицы, просто перечислите имена тех, которые нужны, разделяя запятыми.

```
SELECT firstname, lastname FROM person;
```

Сортировка записей

- По умолчанию, результаты запросов **SELECT** упорядочены так, как они появляются в таблице.
- Если нужно упорядочить результаты особым порядком, например по дате, ID, имени, и т.д., укажите требования в секции **ORDER BY**.

```
SELECT id, firstname, lastname  
FROM person  
ORDER BY lastname;
```


Сортировка по одному столбцу

- По умолчанию ORDER BY сортирует результаты в порядке возрастания (ASC); строки сортируются от А до Z, числа – по возрастанию, даты сортируются от более старых к более новым.
- Можно также указать сортировку по убыванию, используя DESC.

```
SELECT id, firstname, lastname  
FROM person  
ORDER BY lastname DESC;
```

Сортировка по нескольким столбцам

- Можно сортировать не только по одному полю – вы можете указать столько полей, сколько нужно, разделяя запятыми.
- Приоритет сортировки определяется порядком в списке, поэтому если указать ORDER BY lastname, firstname, результаты будут сортироваться по lastname, затем по firstname.

```
SELECT id, firstname, lastname  
FROM person ORDER BY lastname, firstname;
```

Сортировка по позиции столбца

- На столбцы, выбранные для вывода в секциях ORDER BY и GROUP BY можно ссылаться используя их имена, псевдонимы или их позиции.
- Позиции столбцов – целые числа, начиная с 1

```
SELECT id, firstname, lastname  
FROM person ORDER BY 2;
```

Секция WHERE и символы операторов

- Условная секция WHERE используется для **фильтрации** результатов. Используя секцию WHERE можно указать критерии для выбора необходимых записей из таблицы.
- Можно указать одну или более таблиц, разделяя их запятыми для включения их в различные условия в секции WHERE. Однако секция WHERE – необязательная часть команды **SELECT**.
- В секции WHERE можно указать любые условия.
- Можно указать несколько условий, используя AND или OR.
- Секция WHERE также может быть использована с командами SQL **DELETE** или **UPDATE** для указания условия.

WHERE

И СИМВОЛЫ ОПЕРАТОРОВ

Operator	Description	Example
=	Checks if the value of two operands is equal or not, if yes then condition becomes true.	(A = B) is not true.
!=	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

Секция WHERE и символы операторов

```
SELECT * FROM person WHERE `id` != 1;
```

```
SELECT * FROM person WHERE `id` >= 2;
```



Демонстрація



Проверка на NULL

- IS NULL: оператор возвращает true если значение столбца - NULL.
- IS NOT NULL: оператор возвращает true если значение столбца не NULL.

```
SELECT * FROM person WHERE adress IS NULL;  
SELECT * FROM person WHERE adress IS NOT NULL;
```


Секція WHERE и операторы-слова

```
SELECT * FROM person WHERE `id` = 1 OR `id` = 2;
```

```
SELECT * FROM person WHERE `id` = 1 AND `status` = "a";
```

Оператор BETWEEN

- Оператор BETWEEN позволяет указать диапазон для проверки.

```
SELECT  firstname,  
        lastname,  
        age  
FROM    person  
WHERE   age BETWEEN 18 AND 20
```

Оператор IN

- Оператор IN позволяет определить содержится ли указанное значение в списке или в результате подзапроса

```
SELECT column_list  
FROM table_name  
WHERE (expr|column) IN  
( 'value1', 'value2', ... )
```

```
SELECT firstname, lastname, adress  
FROM person  
WHERE adress IN ( 'Kyiv', 'Kiev' )
```

Оператор LIKE

- Оператор MySQL LIKE обычно используется для выбора данных, основанного на совпадении по шаблону.
- Использование оператора LIKE – это способ существенно увеличить производительность запроса.
 - Символ процент (%) позволяет проверить совпадение любой строки с нулем или более символами.
 - Символ подчеркивание (_) позволяет проверить любой один символ.

Оператор LIKE

```
SELECT id, lastName, firstName  
FROM person  
WHERE firstName LIKE 'a%'
```

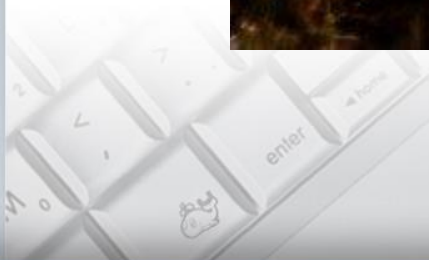
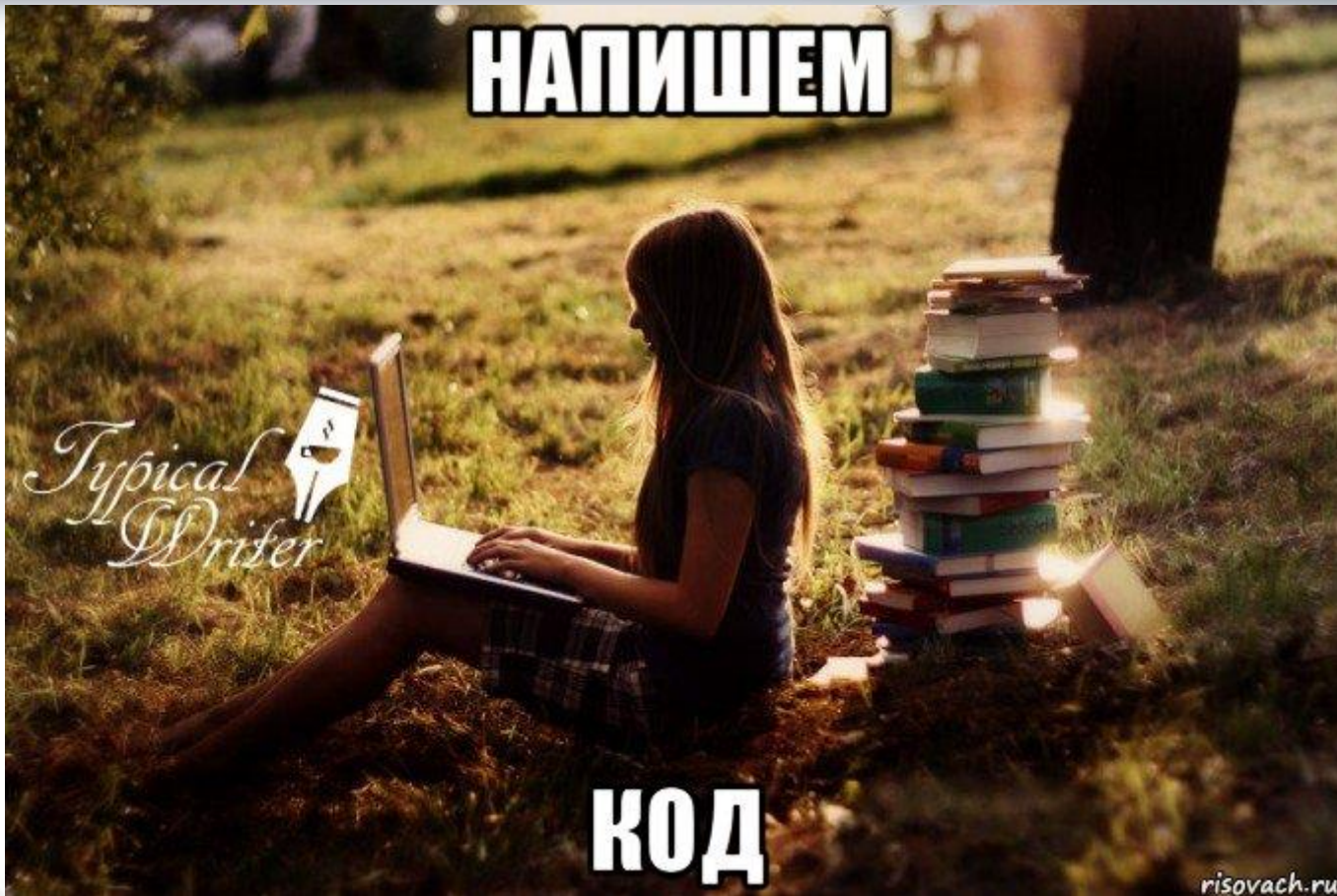
```
SELECT id, lastName, firstName  
FROM person  
WHERE lastname LIKE '%on%'
```

```
SELECT id, lastName, firstName  
FROM person  
WHERE firstname LIKE 'T_m'
```



НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрація



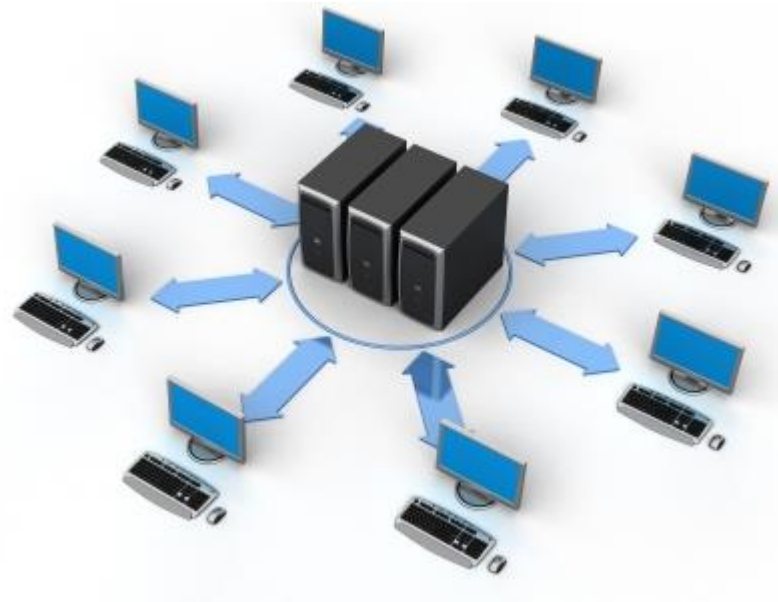


НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА



Технологии распределенных систем и параллельных вычислений

Лекция 1: Основы MySQL



Евгений Беркунский,
кафедра ИУСТ, НУК
<http://www.berkut.mk.ua>
eugeny.berkunsky@gmail.com