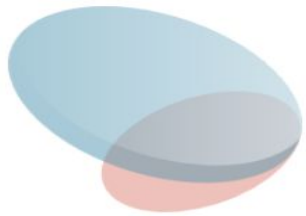


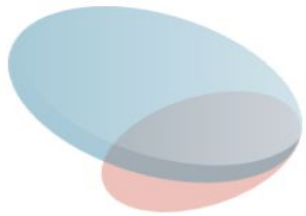
# ОСНОВЫ MySQL

Базы данных.  
MySQL



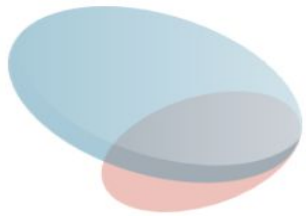
# ОСНОВЫ SQL: MySQL

- Будем использовать MySQL
- СУБД с открытым кодом
- Бесплатная версия (Community Edition) – на [www.mysql.com](http://www.mysql.com)



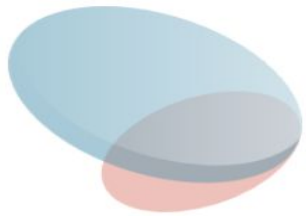
# Основы SQL: языки запросов

- Операции:
  - над множествами (объединение, пересечение, разность, деление, декартово произведение) над отношениями



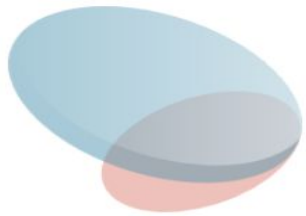
# Основы SQL: языки запросов

- В реляционных СУБД – две группы языков:
  - реляционная алгебра (операнды и результаты – отношения)
  - реляционное исчисление (декларативные языки – запросы)



# ОСНОВЫ SQL: SQL?

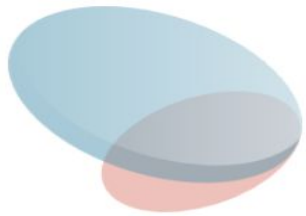
- Structured Query Language — язык структурированных запросов
- Информационно-логический язык, а не язык программирования!
- Для создания, модификации и управления данными в РБД.



# Основы SQL: синтаксис

## Операторы состоят из:

- имен операций и функций;
- имен таблиц и их столбцов;
- зарезервированных ключевых слов и специальных символов;
- логических, арифметических и строковых выражений.



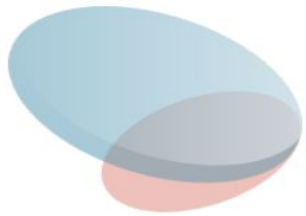
# ОСНОВЫ SQL: синтаксис

Общий вид простого оператора:

**ГЛАГОЛ** параметры;

Пример:

```
SELECT `id` FROM `mytable` ;
```



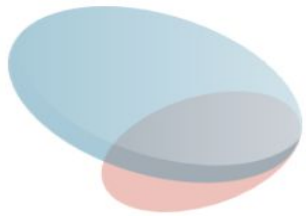
# ОСНОВЫ SQL: синтаксис

- Выражения не зависят от регистра, не требуют наличия кавычек...

- «Правильный» стиль:

```
INSERT INTO `news` (`id`, `post_date`)  
VALUES (42, '2008-06-01 04:13:15');
```





# Основы SQL: синтаксис

Комментарии:

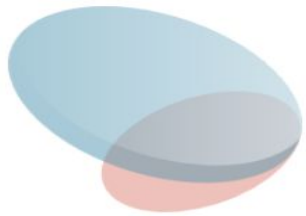
```
-- Структура таблицы
```

```
/*
```

```
Версия сервера: 5.0.54
```

```
Версия PHP: 5.2.6RC1-p11
```

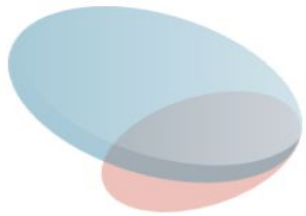
```
*/
```



# Основы SQL: типы данных

Целые числа:

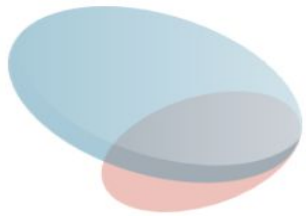
- **TINYINT** — 1 байт, -128..127 (0..255);
- **SMALLINT** — 2 байта, -32768..32767;
- **MEDIUMINT** — 3 байта;
- **INT** — 4 байта;
- **BIGINT** — 8 байт.



# Основы SQL: типы данных

Дробные числа:

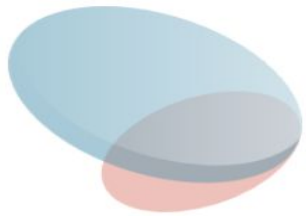
- **FLOAT** (4 байта);
- **DOUBLE** (8 байт).



# Основы SQL: типы данных

Строки:

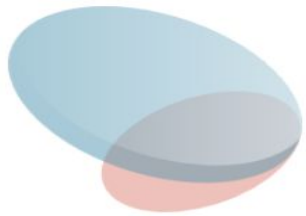
- **CHAR** — дополняет до «ширины»;
- **VARCHAR** — использует минимум;
- **(TINY|MEDIUM|LONG)BLOB** — бинарные данные;
- **TEXT** — текстовые данные;
- **ENUM** — одно из значений;
- **SET** — ноль или более значений.



# ОСНОВЫ SQL: ТИПЫ ДАННЫХ

Другие:

- **BOOL, BOOLEAN;**
- **SERIAL — BIGINT UNSIGNED NOT NULL  
AUTO\_INCREMENT UNIQUE;**
- **DATETIME, DATE, TIMESTAMP, TIME, YEAR.**

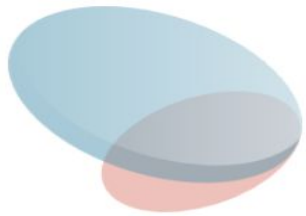


# ОСНОВЫ SQL: КОМАНДЫ

Базы данных: создание

```
CREATE DATABASE `db_name` ;
```

*\* MySQL позволяет создавать БД с кириллическими и спец. символами в названии, но принято использовать латинские буквы, цифры и «\_».*



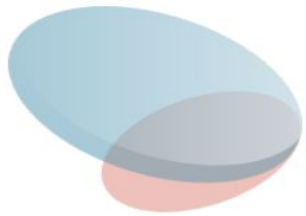
# ОСНОВЫ SQL: КОМАНДЫ

Базы данных: удаление

```
DROP DATABASE `db_name` ;
```

Базы данных: смена текущей

```
USE `db_name` ;
```

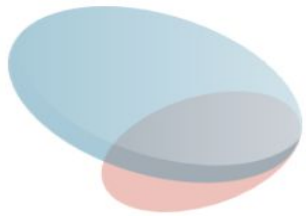


# Основы SQL: команды

Таблицы: создание

```
CREATE TABLE `table_name` (  
    /*описание полей таблицы*/  
    `название_поля` тип параметры,  
    `название_поля` тип,  
    `название_поля` тип параметры  
);
```



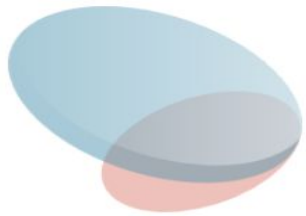


# Основы SQL: команды

Таблицы: создание

Пример:

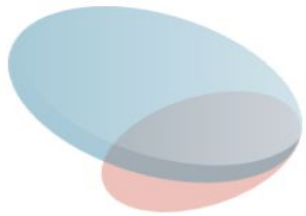
```
CREATE TABLE `news` (  
    `id` mediumint(8) UNSIGNED  
    PRIMARY KEY NOT NULL  
    AUTO_INCREMENT,  
    `posted` TIMESTAMP NOT NULL,  
    `content` TEXT,  
);
```



# ОСНОВЫ SQL: КОМАНДЫ

Таблицы: модификации  
Переименование

```
ALTER TABLE `table_name` RENAME  
TO `table_name2`;
```

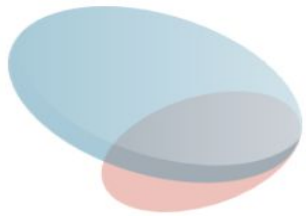


# Основы SQL: команды

Таблицы: модификации

Добавление столбца

```
ALTER TABLE `table_name`  
  ADD COLUMN `new_column`  
  описание_столбца  
  [FIRST | AFTER `after_column`];
```

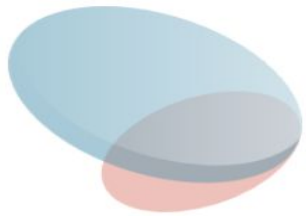


# Основы SQL: команды

Таблицы: модификации

Модификация столбца

```
ALTER TABLE `table_name`  
    MODIFY COLUMN `bad_column`  
    описание_столбца;
```

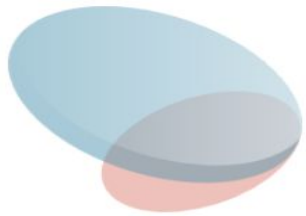


# Основы SQL: команды

Таблицы: модификации

Удаление столбца

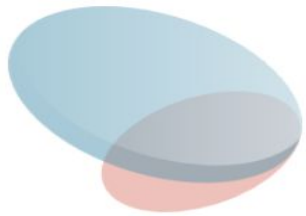
```
ALTER TABLE `table_name`  
    DROP COLUMN  
    `very_bad_column` ;
```



# Основы SQL: команды

Таблицы: удаление

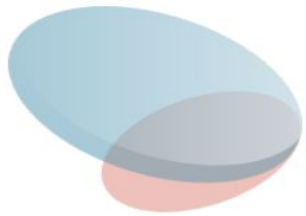
```
DROP TABLE `table_name` ;
```



# ОСНОВЫ SQL: КОМАНДЫ

Строки: добавление

```
INSERT INTO `table_name`  
  [ (`field1`, `field2`, ...,  
  `fieldN`) ]  
VALUES ('value1',  
  'value2', ..., `valueN`);
```

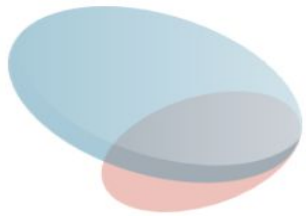


# ОСНОВЫ SQL: КОМАНДЫ

Строки: модификация

```
UPDATE `table_name`  
  SET `field1`='value1',  
      `field2`='value2', ...,  
      `fieldN`='valueN'  
[ WHERE условие ];
```

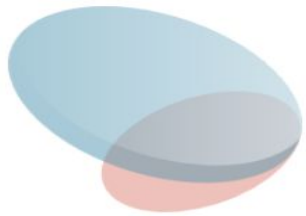




# ОСНОВЫ SQL: КОМАНДЫ

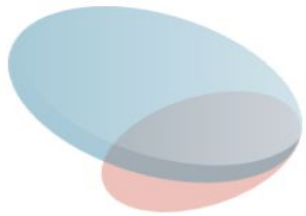
Строки: удаление

```
DELETE FROM `table_name`  
[ WHERE условие ] ;
```



# ОСНОВЫ SQL: ВЫБОРКА

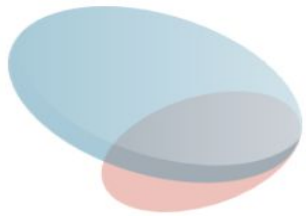
```
SELECT `field1`,  
       `field2`, ..., `fieldN`  
FROM `table1`,  
     `table2`, ..., `tableN`  
[ WHERE условие ]  
[ ORDER BY `field1`,  
           `field2`, ..., `fieldN`  
[ ASC | DESC ] ]
```



# ОСНОВЫ SQL: ВЫБОРКА

```
SELECT * FROM `table_name`;
```

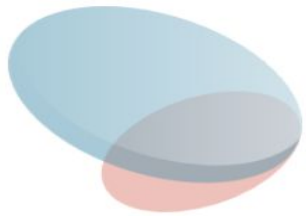
num	id
1	1
1	2
3	2
1	7
2	1
1	42



# ОСНОВЫ SQL: ВЫБОРКА

## WHERE

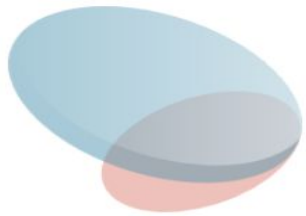
- Условный оператор
- Используется для исключения строк
- Дописывается в выражение с SELECT



# ОСНОВЫ SQL: ВЫБОРКА

```
SELECT * FROM `table_name`  
  WHERE `num` = 1 AND ( `id` > 5  
  OR `id` < 2 );
```

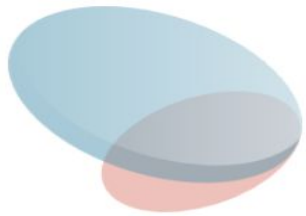
num	id
1	1
1	7
1	42



# ОСНОВЫ SQL: ВЫБОРКА

## **ORDER BY**

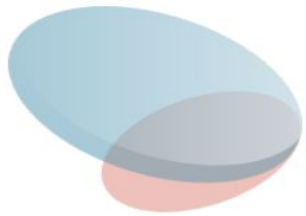
- Определяет порядок сортировки
- Добавляется после WHERE



# ОСНОВЫ SQL: ВЫБОРКА

```
SELECT * FROM `table_name`  
    ORDER BY `num` ASC, `id` DESC;
```

num	id
1	42
1	7
1	2
1	1
2	1
3	2

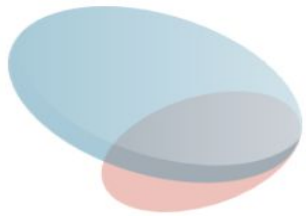


# ОСНОВЫ SQL: ВЫБОРКА

## LIMIT

- Ограничивает число строк
- Добавляется в самый конец

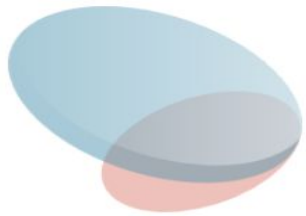




# ОСНОВЫ SQL: ВЫБОРКА

```
SELECT * FROM `table_name`  
LIMIT 2;
```

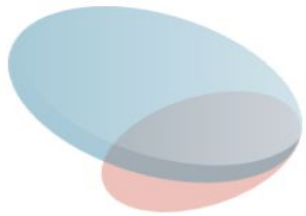
num	id
1	1
1	2



# ОСНОВЫ SQL: ВЫБОРКА

## GROUP BY

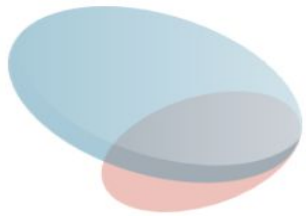
- Группирует строки
- По значениям столбцов
- Каждое значение должно быть  
единственным



# ОСНОВЫ SQL: ВЫБОРКА

```
SELECT * FROM `table_name`  
GROUP BY `num`;
```

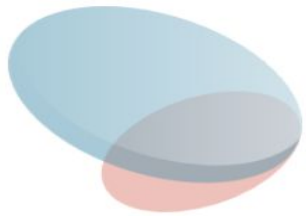
num	id
1	1
2	1
3	2



# ОСНОВЫ SQL: ВЫБОРКА

## HAVING

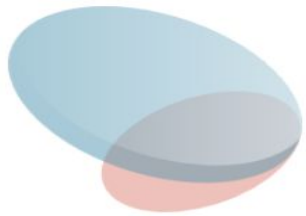
- Аналог WHERE для ORDER BY
- Исключает группы
- Дописывается после ORDER BY



# ОСНОВЫ SQL: ВЫБОРКА

```
SELECT * FROM `table_name`  
    GROUP BY `id` HAVING `id` > 1;
```

num	id
1	2
1	7
1	42

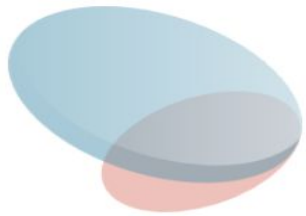


# ОСНОВЫ SQL: функции

SUM – суммирование

```
SELECT SUM(`num`), SUM(`id`) FROM  
`table_name`;
```

SUM(`num`)	SUM(`id`)
9	55

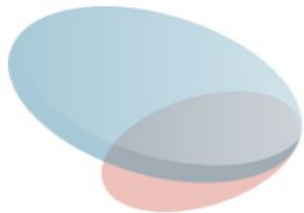


# ОСНОВЫ SQL: функции

COUNT – подсчет строк

```
SELECT COUNT(`num`), COUNT(DISTINCT  
`id`) FROM `table_name`;
```

COUNT(`num`)	COUNT(DISTINCT `num`)
6	3



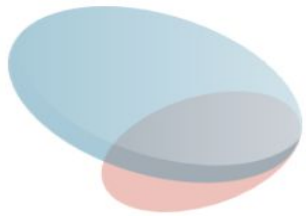
# ОСНОВЫ SQL: ФУНКЦИИ

MIN, MAX – минимум/максимум

```
SELECT MIN(`num`), MAX(`num`)  
FROM `table_name`;
```

MIN(`num`)	MAX(`num`)
1	3



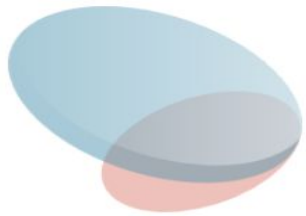


# ОСНОВЫ SQL: функции

AVG – среднее значение

```
SELECT AVG(`num`)  
FROM `table_name`;
```

```
+-----+  
| AVG(`num`) |  
+-----+  
|      1.5000 |  
+-----+
```



# ОСНОВЫ SQL:mysqldump

**mysqldump** позволяет получить снимок содержимого базы данных.

```
mysqldump [OPTIONS] database [tables]
```

**Пример:**

```
mysqldump -u root -p db > laba1.sql
```