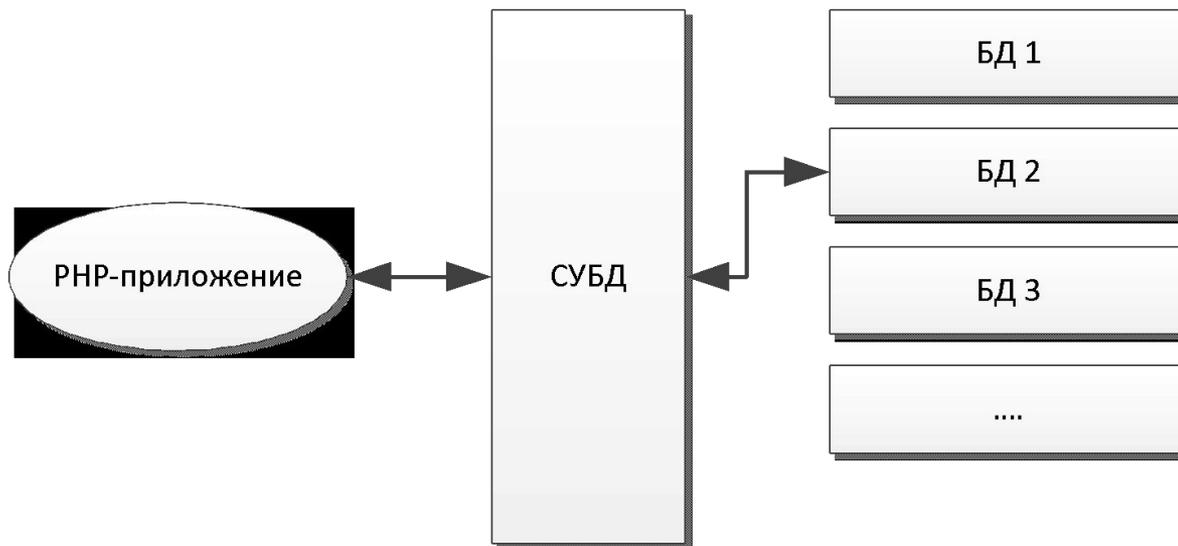


Введение в системы  
управления базами данных  
(СУБД) и основные возможности  
реляционной СУБД MySQL

# Зачем нужна система управления базами данных (СУБД)?

- Быстрый доступ к данным (поиск и сохранение)
- Безопасный одновременный доступ множества пользователей
- Надежность, безопасность, простота использования



# MySQL – самая популярная реляционная СУБД

- Реляционные СУБД используют реляционную модель данных. Проще говоря – данные хранятся в таблицах.
- SQL (structured query language, язык структурированных запросов) – специальный язык для взаимодействия с реляционной СУБД.
- Существуют также нереляционные СУБД (“nosql-базы”) – хранилища «ключ-значение», документо-ориентированные, колоночные, графовые и другие. См., например, Redis, Memcached, MongoDB и т. д.

# Требования к таблицам в реляционных СУБД

- В таблице не должно быть двух одинаковых строк.
- Каждый столбец имеет строго определенный тип данных.
- В каждой ячейке таблицы может находиться только одно значение – так называемое «атомарное» значение (а не группа значений).
- Для идентификации строк и для связи таблиц между собой используется «первичный ключ» – столбец с гарантированно уникальными (в рамках таблицы) значениями.

# Первая нормальная форма



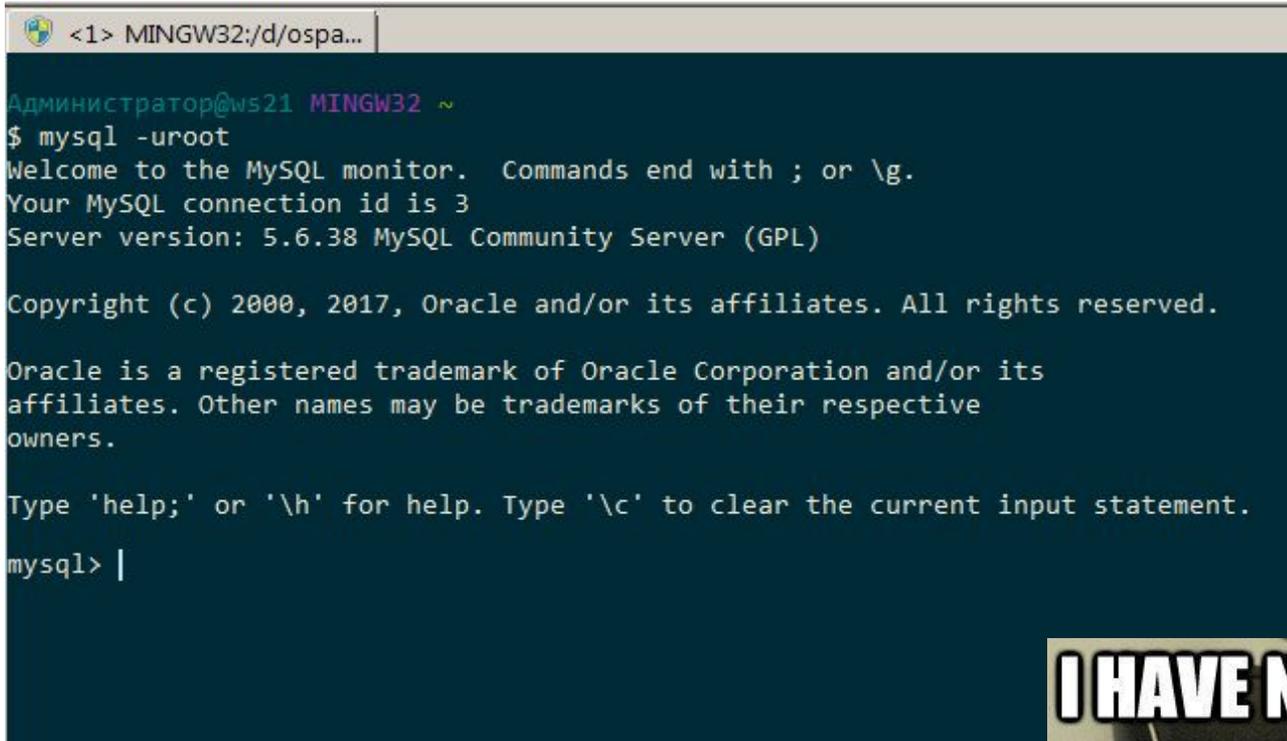
Коты	Мурзик (кот)	4800 г
	Барсик (кот)	5500 г
	Леопольд (кот)	9400 г
	Клякса (кошка)	3600 г
Собаки	Джерри (сука)	12,5 кг
	Шарик (кобель)	10 кг
	Иннокентий (кобель)	28,7 кг
Морские свинки	Кеша (самец)	120
	Чуча (самка)	200



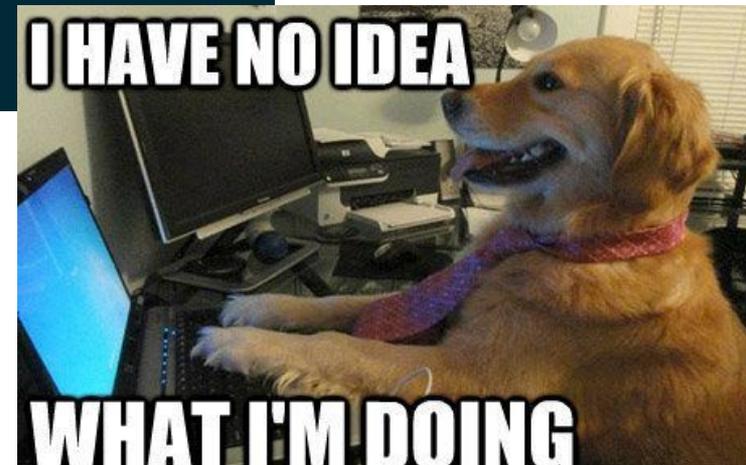
1	Кот	Мурзик	4800	м
2	Кот	Барсик	5500	м
3	Кот	Леопольд	9400	м
4	Кот	Клякса	3600	ж
5	Собака	Джерри	12600	ж
6	Собака	Шарик	10000	м
7	Собака	Иннокентий	28700	м
8	Морская свинка	Кеша	120	м
9	Морская свинка	Чуча	200	ж

# Подключаемся к MySQL через КОНСОЛЬ

```
mysql -uuser -ppassword
```

A screenshot of a terminal window with a dark blue background. The window title bar shows a globe icon and the text "<1> MINGW32:/d/ospa...". The terminal content is as follows:

```
Администратор@ws21 MINGW32 ~  
$ mysql -uroot  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 3  
Server version: 5.6.38 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> |
```



# Первые команды SQL: создаем БД и таблицу

```
CREATE DATABASE
  CHARACTER SET utf8
  COLLATE utf8_general_ci;

USE clinic;

CREATE TABLE `animals` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `species` VARCHAR(50),
  `weight` INT(11),
  `gender` ENUM('м', 'ж'),
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

SHOW TABLES;

DESCRIBE animals;
```

**Обычный пользователь:**



**1980-е**



**Наше время**

# Вставка и выборка данных

```
INSERT INTO animals (name, species, weight, gender)
VALUES
```

```
  ('Мурзик', 'Кот', 4320, 'м'),
  ('Чуча', 'Крыса', 130, 'ж'),
  ('Шарик', 'Собака', 9850, 'м'),
  ('Кеша', 'Попугай', 240, 'м');
```

```
SELECT * FROM animals;
```

```
SELECT * FROM animals ORDER BY weight DESC;
```

```
SELECT * FROM animals WHERE weight > 1000 AND weight < 5000;
```

```
SELECT * FROM animals WHERE name = 'мурзик'
  OR NAME LIKE 'шар%' OR NAME LIKE '%ча' OR NAME LIKE '%р%';
```

# Изменение и удаление данных



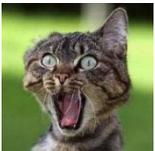
```
UPDATE animals SET name = 'Мурзик', weight = 3480;
```



```
UPDATE animals SET name = 'Бобик', weight = 2000  
WHERE id = 3;
```



```
UPDATE animals SET name = 'Пушинка'  
WHERE weight < 100;
```

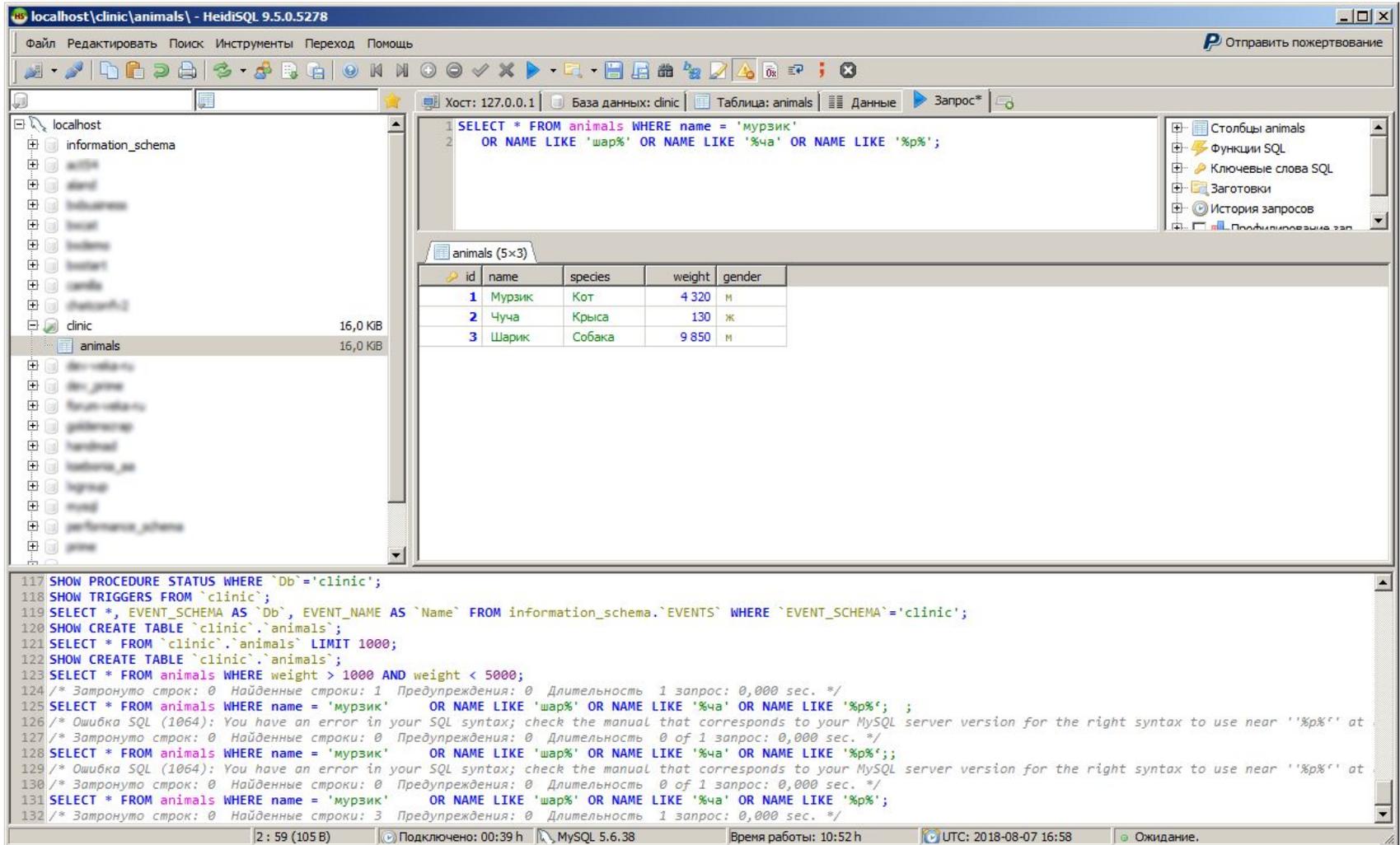


```
DELETE FROM animals WHERE weight > 10.000;
```



```
DELETE FROM animals WHERE id = 3;
```

# HeidiSQL



The screenshot shows the HeidiSQL interface with the following components:

- Left Panel:** A tree view of the database structure. The 'clinic' database is selected, and the 'animals' table is highlighted. The size of the 'animals' table is 16,0 KIB.
- Top Panel:** The SQL editor contains the following query:

```
1 SELECT * FROM animals WHERE name = 'мурзик'
2   OR NAME LIKE 'шар%' OR NAME LIKE '%ча' OR NAME LIKE '%р%';
```
- Right Panel:** A list of table columns for 'animals', including 'id', 'name', 'species', 'weight', and 'gender'. Other options like 'Функции SQL' and 'Ключевые слова SQL' are also visible.
- Table View:** Below the query editor, a table titled 'animals (5x3)' displays the results of the query:

id	name	species	weight	gender
1	Мурзик	Кот	4 320	м
2	Чуча	Крыса	130	ж
3	Шарик	Собака	9 850	м
- Bottom Panel:** A log window showing the execution of various SQL commands, including 'SHOW PROCEDURE STATUS', 'SHOW TRIGGERS', and the execution of the query above. It includes status messages like 'Найденные строки: 3' and 'Длительность 1 запрос: 0,000 sec.'.

# Бэкап и восстановление

Выгрузка БД в текстовый файл:

```
mysqldump -uuser -ppassword clinic > clinic.sql
```

Загрузка БД из текстового файла:

```
mysql -uuser -ppassword clinic < clinic.sql
```



# Группировка

Хост: 127.0.0.1 | База данных: clinic | Таблица: animals | Данные | Запрос\*

```
1 SELECT *, COUNT(*) as total FROM animals
2   GROUP BY species ORDER BY total;
3
```

animals (6×4)

id	name	species	weight	gender	total
4	Кеша	Попугай	240	м	1
3	Шарик	Собака	9 850	м	2
2	Чуча	Крыса	130	ж	2
1	Мурзик	Кот	4 320	м	3

Хост: 127.0.0.1 | База данных: clinic | Запрос\*

```
1 SELECT species, COUNT(species) as total FROM animals
2   GROUP BY species ORDER BY total;
3
```

animals (2×4)

species	total
Попугай	1
Собака	2
Крыса	2
Кот	3

# Агрегирующие функции

- Функции: COUNT, MAX, MIN, SUM, AVG
- Могут использоваться как вместе с GROUP BY, так и отдельно

```
SELECT species, SUM(weight) AS total  
FROM animals  
GROUP BY species  
ORDER BY total;
```

```
SELECT MAX(weight) FROM animals;
```