

## Запросы из нескольких таблиц

Информатика  
10-11 классы

4 декабря 2012 г.

# Повторим: GROUP BY и HAVING

id	student_id	gender	mark	created_at
----	------------	--------	------	------------

Таблица: exam\_results

- Список учеников, у которых средний балл выше 3? При этом отсортировать по убыванию баллов?

## Listing 1: GROUP BY HAVING

```
SELECT *, AVG(mark) AS avg_mark FROM exam_results
GROUP BY student_id
HAVING avg_mark
ORDER BY avg_mark DESC;
```

id	student_id	gender	mark	created_at
----	------------	--------	------	------------

Таблица: exam\_results

Найти сумму всех положительных оценок  
учеников

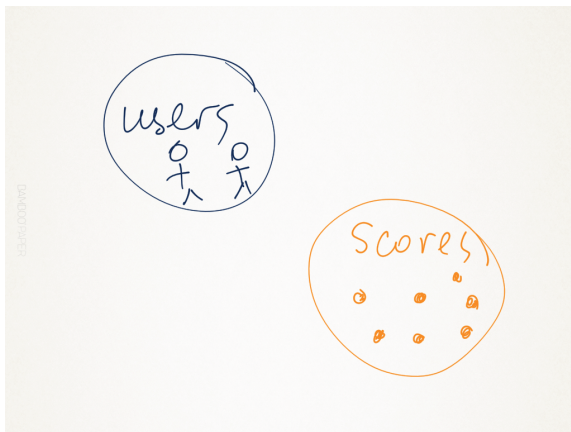
id	student_id	gender	mark	created_at
----	------------	--------	------	------------

Таблица: exam\_results

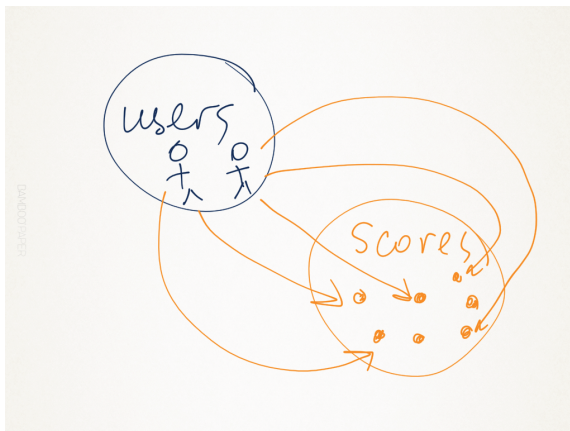
## Listing 2: SUM

```
SELECT SUM(mark) AS sum FROM exam_results
WHERE mark > 2;
```

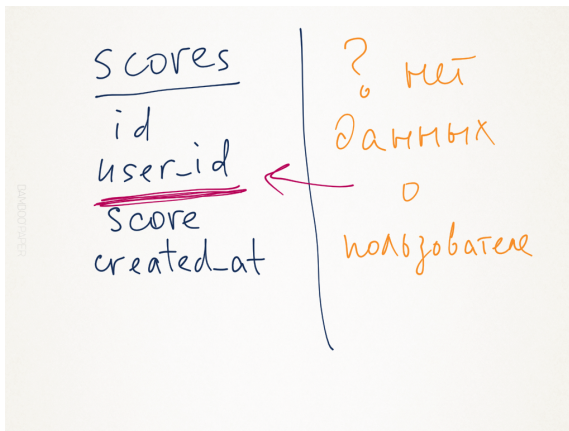
# Постановка



# Постановка



# Постановка



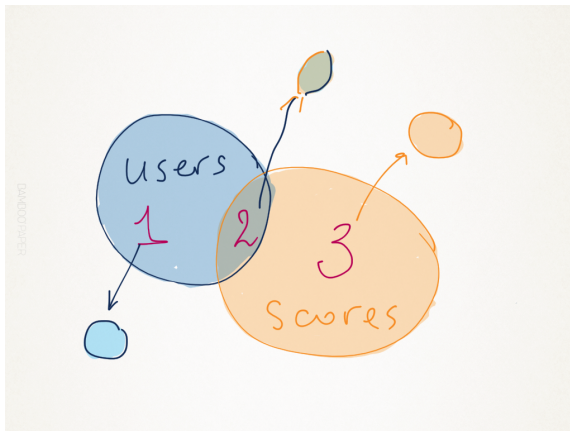
Как получить данные о десяти пользователях,  
набравших большинство очков?



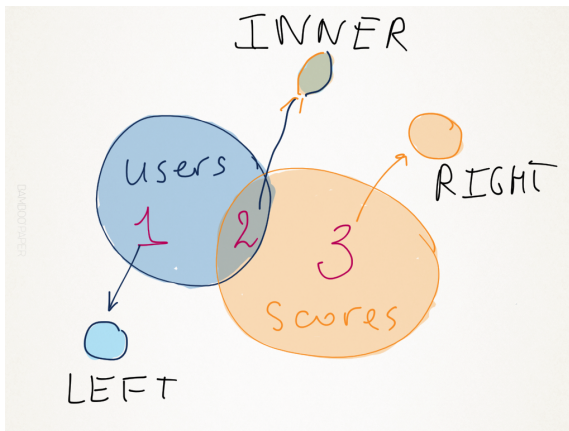
# Оператор JOIN

- Оператор **JOIN** позволяет объединить две таблицы в выдаче.
- Запрос звучит следующим образом: ИЗВЛЕЧЬ все записи из таблицы пользователей и добавить к ним записи из таблицы баллов, при этом для каждого пользователя прикреплять только его баллы.
- Запрос JOIN всегда возвращает множество пар “пользователи–баллы”.

## JOIN



# Виды JOIN



# Пример LEFT JOIN

id	first_name	last_name	gender	created_at
----	------------	-----------	--------	------------

Таблица: users

id	user_id	score	created_at
----	---------	-------	------------

Таблица: scores

- Пример запроса с JOIN'ом

## Listing 3: LEFT JOIN

```
SELECT * FROM users u
LEFT JOIN scores s
ON u.id = s.user_id;
```

# Что выведет LEFT JOIN

id	first_name	last_name	gender	created_at
1	Vasya	Ivanov	1	2012-01-01
2	Masha	Petrova	2	2012-03-01
3	Kos	Palpatine	1	2012-04-08

Таблица: users

id	user_id	score	created_at
1	1	100	2012-01-01
2	1	50	2012-01-05
3	2	195	2012-03-03
4	1	5	2012-03-12
5	4	205	2012-04-12

Таблица: scores

- В результате будет 4 записи.
- Три — для Васи (связи Вася — первый балл, Вася — второй балл, Вася — третий балл) и одна — для Маши (Маша — её первый балл).

# Задача на сумму

id	first_name	last_name	gender	created_at
----	------------	-----------	--------	------------

id	user_id	score	created_at
----	---------	-------	------------

- Посчитаем, сколько баллов набрал каждый пользователь, и выведем на экран таблицу в формате id, имя, фамилия, сумма очков.

## Listing 4: LEFT JOIN

```
SELECT u.id, u.first_name, u.last_name, SUM(s.score)
FROM users u
LEFT JOIN scores s
ON u.id = s.user_id;
```

Почему предыдущий запрос  
работает не так, как хочется?



# Правильный вариант LEFT JOIN

- Мы получили одну запись, вместо простого списка пользователей.
- Хотелось бы сгруппировать результаты (привет, GROUP BY).

## Listing 5: LEFT JOIN GROUP BY

```
SELECT u.id , u.first_name , u.last_name , SUM(s.score)
FROM users u
LEFT JOIN scores s
ON u.id = s.user_id
GROUP BY u.id ;
```

Вопрос: что сделает LEFT JOIN, если у пользователя нет баллов?

# Таблицы для задачи

id	side_id	first_name	last_name
1	1	Mace	Windu
2	2	Kos	Palpatine
3	1	Yoda	
4	2	Dooku	
5	1	Obi-Wan	Kenobi

Таблица: persons

id	side
1	Jedi
2	Sith

Таблица: sides