

# Задача №2

## Условие:

Написать SQL-запрос. Вывести данные по ученикам в формате: age, gender, is\_excellent, total. Отсортировать по age и is\_excellent в обратном порядке, вывести только если total > 1.

Здесь:

is\_excellent - является ли отличником (0 - не является, 1 - является);

total - количество записей (выводить при значении 1 и более).

```
CREATE TABLE `users` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `age` int(11) DEFAULT NULL,  
  `gender` tinyint(1) DEFAULT '0' COMMENT '0 - male, 1 - female',  
  `mark` int(11) DEFAULT NULL COMMENT '1-5',  
  PRIMARY KEY (`id`)  
);
```

## Решение:

В данном случае БД не нормализована, информация об учениках и их оценках хранится в одной таблице, при этом в условии задачи требуется выводить вычисляемое поле total, что не имеет смысла, поскольку в таблице users каждая запись ученика уникальна (первичный ключ ID), и поле total будет всегда содержать 1. Тем не менее для выборки данных из представленной таблицы, учитывая ряд заданных условий, и полагая, что значение поля mark равное NULL является признаком нулевого кол-ва записей (total = 0), можно воспользоваться SQL-запросом (листинг 2)<sup>1</sup>:

Листинг 2 – SQL-запрос выборки данных

```
SELECT  
  t.age,  
  t.gender,
```

---

<sup>1</sup> В качестве СУБД при решении задачи использовалась MySQL-5.6

```

t.mark,
IF(t.mark = 5, 1, 0) as is_excellent,
1 as total
FROM
`users-old` as t
WHERE
t.mark IS NOT NULL
ORDER BY
t.`age` DESC, `is_excellent` DESC

```

Результат выполнения SQL-запроса в среде PHPMyAdmin представлен на рисунке 2:

age ▼ 1	gender 0 - male, 1 - female	is_excellent ▼ 2
33	0	1
32	0	1
30	1	1
30	1	0
29	1	0
26	1	0
22	1	1
22	0	0
20	1	1
18	1	0

Рисунок 1 – Результат выполнения SQL-запроса в PHPMyAdmin

Однако можно рассмотреть вариант с нормализацией представленной БД, и разбиением таблицы на несколько связанных между собой таблиц. Для нормализации, в самом простом случае, можно разбить данную таблицу на 2 таблицы, следующим образом (таблица 1):

Таблица 1 – Список таблиц БД после нормализации

Имя таблицы	Описание
users	<p>Содержит информацию о пользователях:</p> <ul style="list-style-type: none"> <li>• идентификатор пользователя (первичный ключ);</li> <li>• возраст;</li> <li>• пол.</li> </ul>

users_marks	<p>Содержит информацию об оценках каждого пользователя:</p> <ul style="list-style-type: none"> <li>• идентификатор записи (первичный ключ);</li> <li>• идентификатор пользователя (внешний ключ);</li> <li>• оценка;</li> <li>• временная метка выставления оценки.</li> </ul>
-------------	--

Схема новой БД будет выглядеть следующим образом (рисунок 3):



Рисунок 2 – Схема нормализованной БД в РНРMyAdmin

Как можно заметить, таблица users связана отношением один-ко-многим с таблицей users\_marks.

SQL-дамп нормализованной БД представлен в листинге 1:

```
--
-- Структура таблицы `users`
--

CREATE TABLE `users` (
  `user_id` int(11) UNSIGNED NOT NULL,
  `age` int(11) DEFAULT NULL,
  `gender` tinyint(1) DEFAULT '0' COMMENT '0 - male, 1 - female'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `users`
--

INSERT INTO `users` (`user_id`, `age`, `gender`) VALUES
(1, 18, 1),
(2, 20, 1),
```

```
(3, 22, 0),
(4, 22, 1),
(5, 30, 1),
(6, 32, 0),
(7, 30, 1),
(8, 33, 0),
(9, 26, 1),
(10, 29, 1),
(11, 18, 0);
```

```
-- -----
```

```
--
```

```
-- Структура таблицы `users_marks`
```

```
--
```

```
CREATE TABLE `users_marks` (
  `id` int(11) UNSIGNED NOT NULL,
  `user_id` int(11) UNSIGNED NOT NULL,
  `mark` int(11) UNSIGNED NOT NULL COMMENT '1-5',
  `tms` timestamp NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
```

```
-- Дамп данных таблицы `users_marks`
```

```
--
```

```
INSERT INTO `users_marks` (`id`, `user_id`, `mark`, `tms`) VALUES
(1, 1, 5, '2019-03-28 14:20:23'),
(2, 2, 3, '2019-03-28 14:20:23'),
(3, 3, 4, '2019-03-28 14:20:23'),
(4, 4, 4, '2019-03-28 14:20:23'),
(5, 5, 3, '2019-03-28 14:20:23'),
(6, 6, 4, '2019-03-28 14:20:23'),
(7, 7, 5, '2019-03-28 14:20:23'),
(8, 8, 2, '2019-03-28 14:20:23'),
(9, 9, 5, '2019-03-28 14:20:23'),
(10, 10, 4, '2019-03-28 14:20:23'),
(11, 1, 5, '2019-03-28 14:20:23'),
(12, 5, 4, '2019-03-28 14:20:23');
```

```
--
```

```
-- Индексы сохранённых таблиц
```

```
--
```

```

--
-- Индексы таблицы `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`);

--
-- Индексы таблицы `users_marks`
--
ALTER TABLE `users_marks`
  ADD PRIMARY KEY (`id`),
  ADD KEY `FK_users` (`user_id`);

--
-- AUTO_INCREMENT для сохранённых таблиц
--

--
-- AUTO_INCREMENT для таблицы `users`
--
ALTER TABLE `users`
  MODIFY `user_id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;

--
-- AUTO_INCREMENT для таблицы `users_marks`
--
ALTER TABLE `users_marks`
  MODIFY `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;

--
-- Ограничения внешнего ключа сохраненных таблиц
--

--
-- Ограничения внешнего ключа таблицы `users_marks`
--
ALTER TABLE `users_marks`
  ADD CONSTRAINT `FK_users` FOREIGN KEY (`user_id`) REFERENCES `users`
  (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

```

**Заполним тестовыми данными созданные таблицы (рисунки 4-5):**

+ Параметры

				user_id	age	gender 0 - male, 1 - female
<input type="checkbox"/>				1	18	1
<input type="checkbox"/>				2	20	1
<input type="checkbox"/>				3	22	0
<input type="checkbox"/>				4	22	1
<input type="checkbox"/>				5	30	1
<input type="checkbox"/>				6	32	0
<input type="checkbox"/>				7	30	1
<input type="checkbox"/>				8	33	0
<input type="checkbox"/>				9	26	1
<input type="checkbox"/>				10	29	1
<input type="checkbox"/>				11	18	0

☐ Отметить все
 С отмеченными:

Рисунок 3 – Данные таблицы users в RHPMyAdmin

				id	user_id	mark 1-5	tms
<input type="checkbox"/>				1	1	5	2019-03-28 19:20:23
<input type="checkbox"/>				2	2	3	2019-03-28 19:20:23
<input type="checkbox"/>				3	3	4	2019-03-28 19:20:23
<input type="checkbox"/>				4	4	4	2019-03-28 19:20:23
<input type="checkbox"/>				5	5	3	2019-03-28 19:20:23
<input type="checkbox"/>				6	6	4	2019-03-28 19:20:23
<input type="checkbox"/>				7	7	5	2019-03-28 19:20:23
<input type="checkbox"/>				8	8	2	2019-03-28 19:20:23
<input type="checkbox"/>				9	9	5	2019-03-28 19:20:23
<input type="checkbox"/>				10	10	4	2019-03-28 19:20:23
<input type="checkbox"/>				11	1	5	2019-03-28 19:20:23
<input type="checkbox"/>				12	5	4	2019-03-28 19:20:23

☐ Отметить все
 С отмеченными:

Рисунок 4 – Данные таблицы users\_marks в RHPMyAdmin

После нормализации, можно сформировать запрос на выборку записей с указанными в условии задачи критериями, при этом также можно учесть число имеющихся оценок (total) для каждого пользователя (число записей для пользователя в таблице users\_marks), и если  $total \geq 1$ , то выводить информацию по данному пользователю (INNER JOIN).

Пример кода рассматриваемого SQL-запроса представлен в листинге 3:

### Листинг 3 – SQL-запрос выборки данных для нормализованный БД

```
SELECT
-- идентификатор ученика (выборка уникальных значений)
DISTINCT u.user_id,
-- возраст ученика
u.age,
-- пол ученика
u.gender,
-- признак того, что ученик отличник - подзапрос, подсчитывающий число записей
из таблицы оценок для данного пользователя, у которых оценка не равна 5, т.е.
для отличников данный подзапрос вернет 0
IF((SELECT COUNT(id) FROM users_marks WHERE user_id = u.user_id AND mark != 5)
= 0, 1, 0) is_excellent,
-- кол-во записей для конкретного ученика в таблице оценок users_marks
(SELECT COUNT(id) FROM users_marks WHERE user_id = u.user_id) total
FROM
users u
-- внутренний join с таблицей оценок по полю user_id
INNER JOIN users_marks um ON um.user_id = u.user_id
-- сортировка по убыванию для полей age, is_excellent
ORDER BY
u.age DESC, is_excellent DESC
```

Результат выполнения данного SQL-запроса в среде PHPMыAdmin представлен на рисунке 6.

user_id	age ▾ 1	gender	is_excellent ▾ 2	total
8	33	0	0	1
6	32	0	0	1
7	30	1	1	1
5	30	1	0	2
10	29	1	0	1
9	26	1	1	1
3	22	0	0	1
4	22	1	0	1
2	20	1	0	1
1	18	1	1	2

Рисунок 5 - Результат выполнения SQL-запроса на выборку из нескольких таблиц в PHPMыAdmin

Как можно заметить, записи для пользователя с `user_id = 11` нет в выборке, поскольку для данного пользователя нет записей в таблице оценок `users_marks`, что позволяет выбирать только те записи, у которых `total >= 1`.

Также полученный запрос можно инкапсулировать в представление для дальнейшего использования (например, материализованное представление для ускорения выполнения запроса).