

Базы данных. SQL. Вложенные запросы.



Окулов Антон

R.class

ООО “Авто-плюс”

id	name	role	salary	department_id
1	Иванов Семен	Директор	100000	1
2	Петрова Светлана	Бухгалтер	50000	1
3	Жуков Роман	Разнорабочий	15000	2
4	Рогов Дмитрий	Программист	20000	3
5	Баранов Петр	Автомеханик	22000	4
6	Мигунов Олег	Разнорабочий	16000	2
7	Глазунов Аркадий	Автомеханик	21000	4

users

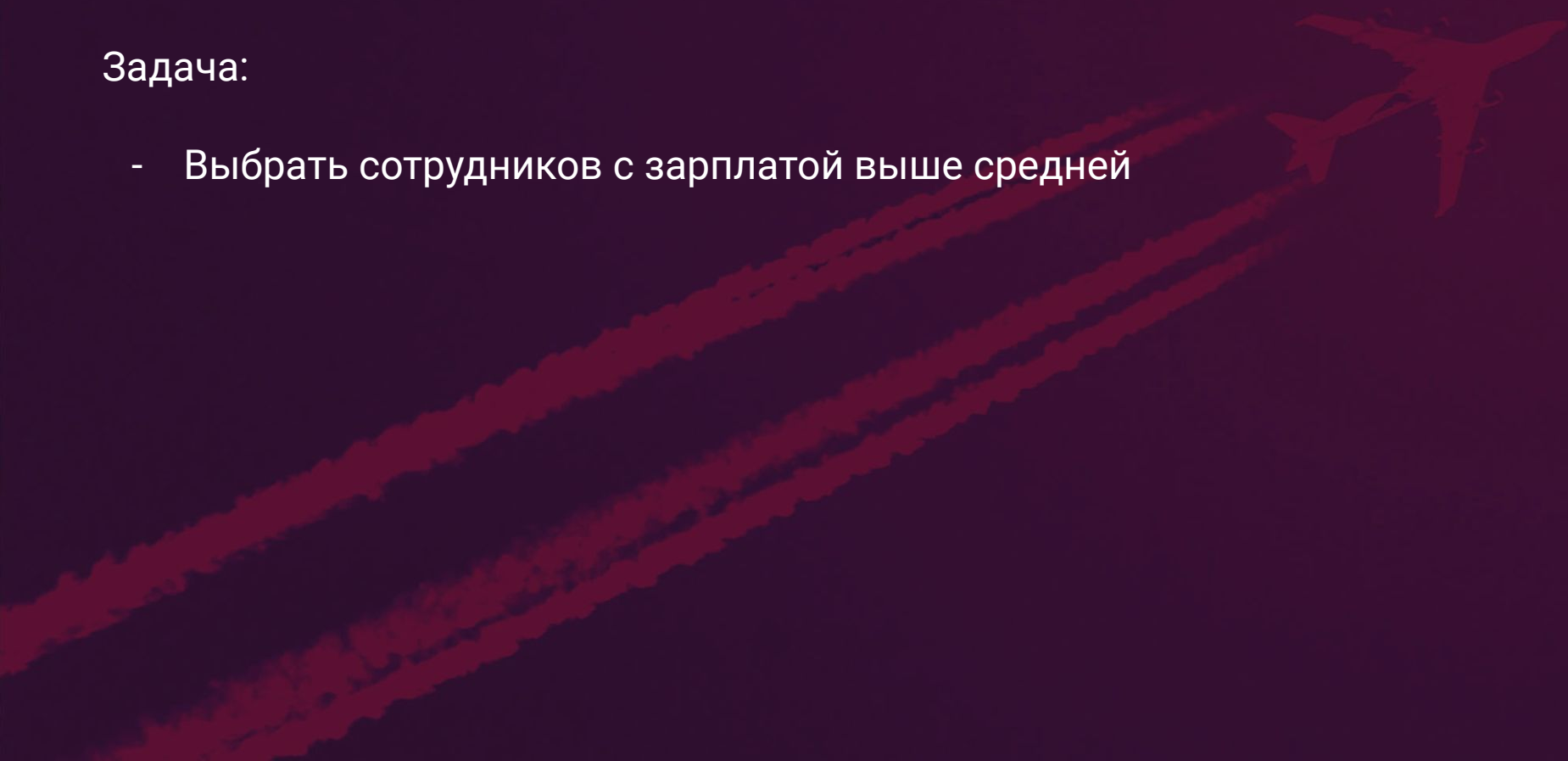
id	name
1	Администрация
2	Рабочие
3	Программисты
4	Механики

departments

Подзапрос как скалярный операнд

Задача:

- Выбрать сотрудников с зарплатой выше средней



Подзапрос как скалярный операнд

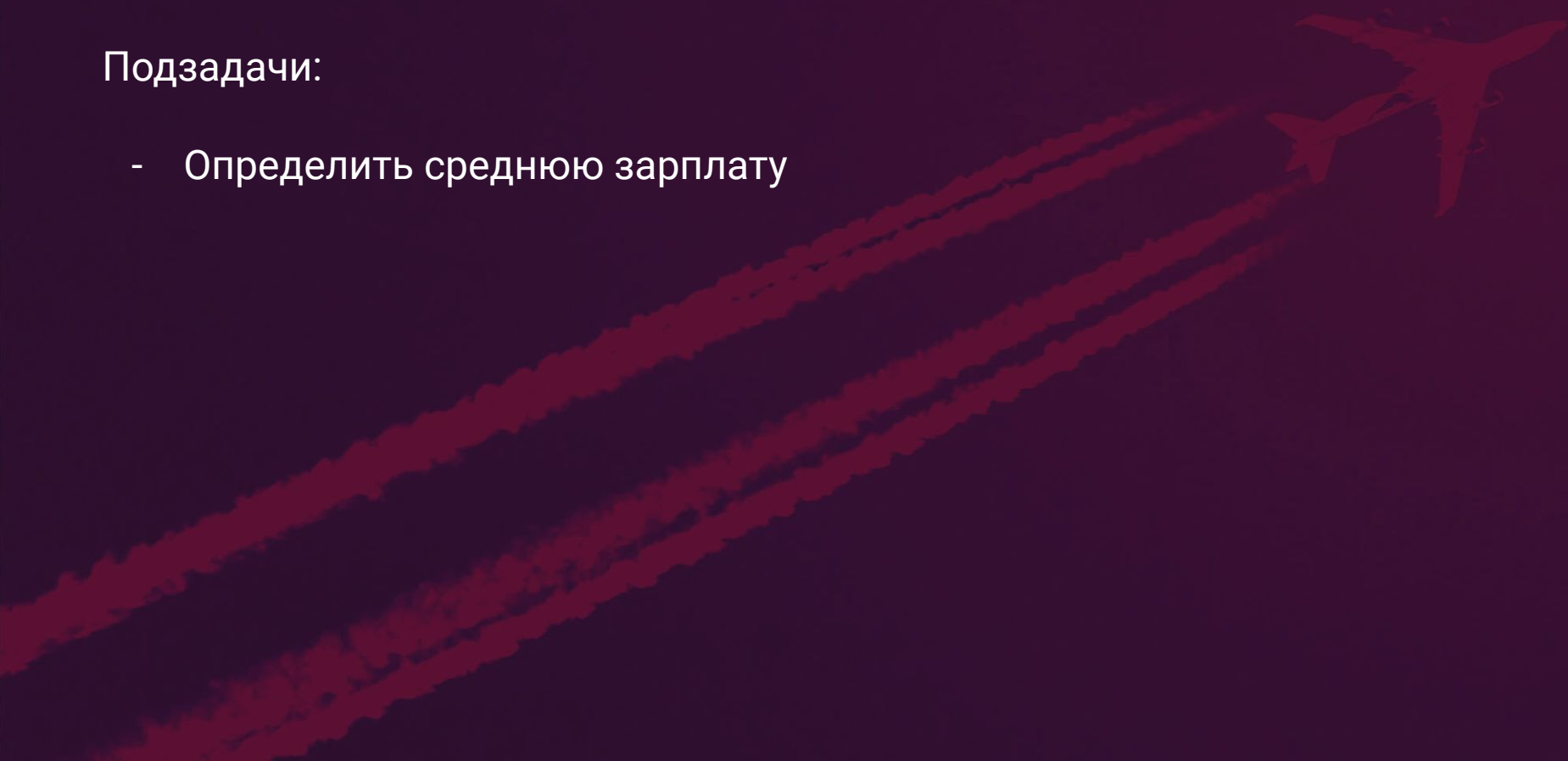
Подзадачи:



Подзапрос как скалярный операнд

Подзадачи:

- Определить среднюю зарплату



Подзапрос как скалярный операнд

Подзадачи:

- Определить среднюю зарплату
- Выбрать сотрудников с зарплатой выше средней



Средняя зарплата

```
SELECT AVG(`salary`) FROM `workers`
```

AVG(`salary`)
34857.1429

Сотрудники

```
SELECT * FROM
  `workers`
WHERE
  `salary` > (SELECT AVG(`salary`) FROM `workers`)
```

id	name	role	salary	department_id
1	Иванов Семен	Директор	100000	1
2	Петрова Светлана	Бухгалтер	50000	1

Подзапрос как скалярный операнд

Задача:

- Выбрать всех сотрудников из отдела в котором работает самый высокооплачиваемый сотрудник



Подзапрос как скалярный операнд

Подзадачи:

- Определить максимальную зарплату
- Выбрать ид отдела для сотрудника с максимальной зарплатой
- Выбрать сотрудников по ид отдела



Максимальная зарплата

```
SELECT MAX(`salary`) FROM `workers`
```

MAX(`salary`)
100000

ИД отдела с максимальной зарплатой

```
SELECT `department_id` FROM `workers` WHERE `salary` = (  
  SELECT MAX(`salary`) FROM `workers`  
)
```

department_id
1

Выбрать всех сотрудников из отдела

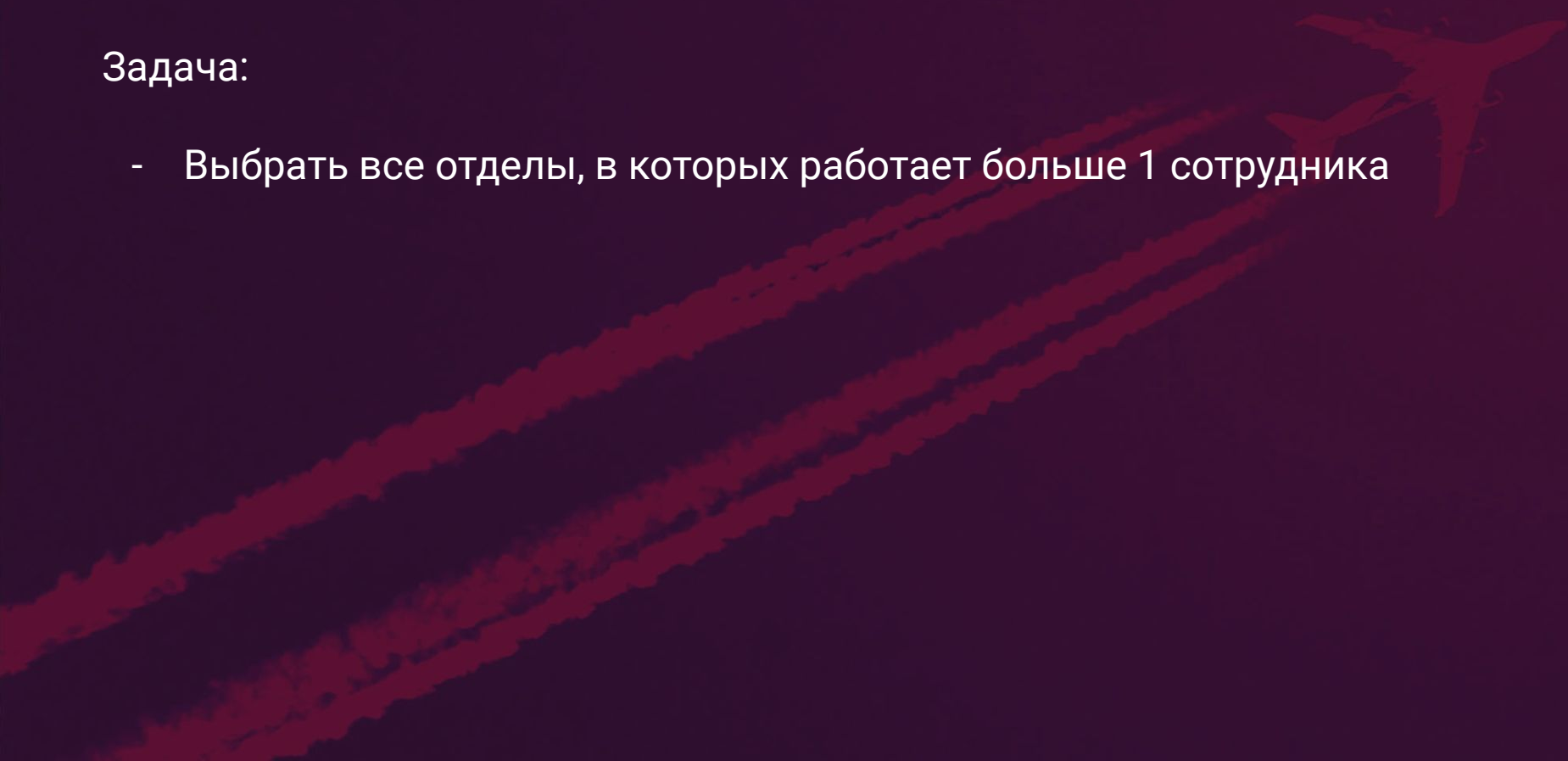
```
SELECT * FROM `workers` WHERE `department_id` = (  
  SELECT `department_id` FROM `workers` WHERE `salary` = (  
    SELECT MAX(`salary`) FROM `workers`  
  )  
)
```

id	name	role	salary	department_id
1	Иванов Семен	Директор	100000	1
2	Петрова Светлана	Бухгалтер	50000	1

Подзапросы, возвращающие несколько строк

Задача:

- Выбрать все отделы, в которых работает больше 1 сотрудника



Подзапросы, возвращающие несколько строк

Подзадачи:

- Выбрать ид отделов с количеством сотрудников больше 1
- По заданному список ид отделов выбрать сами отделы



Выбираем ид отделов

```
SELECT `department_id` FROM `workers`  
GROUP BY `department_id` HAVING COUNT(*) > 1
```

department_id
1
2
4

Выбираем отделы

```
SELECT * FROM `departments` WHERE `id` IN (  
    SELECT `department_id` FROM `workers`  
    GROUP BY `department_id` HAVING COUNT(*) > 1  
)
```

id	name
1	Администрация
2	Рабочие
4	Механики

Выбираем отделы

```
SELECT * FROM `departments` WHERE `id` IN (  
    SELECT `department_id` FROM `workers`  
    GROUP BY `department_id` HAVING COUNT(*) > 1  
)
```

id	name
1	Администрация
2	Рабочие
4	Механики

Инвертируем условие

```
SELECT * FROM `departments` WHERE `id` NOT IN (  
    SELECT `department_id` FROM `workers`  
    GROUP BY `department_id` HAVING COUNT(*) > 1  
)
```

id	name
3	Программисты

Ключевые слова ANY, ALL

Для работы с условиями $>$, $<$, $>=$, $<=$, $<>$, $=$ оператора IN / NOT IN недостаточно.

Для этого применяются ключевые слова ANY (SOME) и ALL

Вид	Описание	Применение
ANY	Под условие подходит любое значение	<code>`user_id` > ANY (SELECT ...)</code>
ALL	Под условие подходят все значения	<code>`user_id` <= ALL (SELECT ...)</code>

Коррелированные подзапросы

Задача:

- Выбрать отделы с указанием средней зарплаты в каждом из них



Коррелированные подзапросы

```
SELECT *, (  
    SELECT AVG(`salary`) FROM `workers`  
    WHERE `workers`.`department_id` = `departments`.`id`  
) as `avg_salary` FROM `departments`
```

id	name	avg_salary
1	Администрация	75000.0000
2	Рабочие	15500.0000
3	Программисты	20000.0000
4	Механики	21500.0000

Конструкция EXISTS

```
SELECT * FROM `departments` WHERE EXISTS (  
  SELECT * FROM `workers`  
  WHERE  
    `workers`.`department_id` = `departments`.`id` AND  
    `salary` > 20000  
)
```

id	name
1	Администрация
4	Механики

Подзапросы в секции FROM / JOIN

```
SELECT *  
FROM  
  departments,  
  (  
    SELECT  
      `department_id`,  
      MIN(`salary`) as `min_salary`  
    FROM `workers`  
    GROUP BY `department_id`  
  ) `min_salaries`  
WHERE  
  `min_salaries`.`department_id` = `departments`.`id`
```


Подзапросы в секции FROM / JOIN

```
SELECT *  
FROM  
  departments,  
  (  
    SELECT  
      `department_id`,  
      MIN(`salary`) as `min_salary`  
    FROM `workers`
```

id	name	department_id	min_salary
1	Администрация	1	50000
2	Рабочие	2	15000
3	Программисты	3	20000
4	Механики	4	21000

```
ents`.`id`
```

INSERT + SELECT

```
INSERT INTO `old_workers`  
VALUES (`name`, `old_salary`)  
SELECT  
  `workers`.`name`,  
  `workers`.`salary`  
FROM `workers`  
WHERE `workers`.`id` IN (1,2,3)
```

Ограничения подзапросов

- Нельзя изменять (удалять/изменять) данные в таблице, если ее данные используются в подзапросе этого запроса

```
DELETE FROM t WHERE ... (SELECT ... FROM t ...);  
UPDATE t ... WHERE col = (SELECT ... FROM t ...);  
{INSERT|REPLACE} INTO t (SELECT ... FROM t ...);
```

- Не поддерживается LIMIT в подзапросах

Подробнее и больше об ограничениях:

<https://dev.mysql.com/doc/mysql-reslimits-excerpt/8.0/en/subquery-restrictions.html>

Производительность

- Если есть возможность избежать подзапроса - его необходимо избежать
- Стараться использовать как можно меньше уровней вложенности подзапросов

Bcë :)

