Приложение 1

|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ ФГБОУ ВПО “Ульяновский государственный университет” | Форма |  |
| Ф-Титульный лист лабораторной работы |  |  |

Министерства науки и высшего образования РФ Ульяновский государственный университет

Факультет Математики, информационных и авиационных технологий Кафедра Телекоммуникационные технологии и сети

ЛАБОРАТОРНАЯ РАБОТА

по дисциплине

Базы данных

(название дисциплины)

SQL. Запросы

(название темы)

Направление бакалавриата Информационные системы и технологии. 09.03.02

Работу выполнил студент ИС-З-21 31.10.2022 Гусев В.Е.

(группа) (подпись, дата) (Ф.И.О.)

Научный руководитель

(должность) (подпись, дата) (Ф.И.О.)

(оценка)

У Л Ь Я Н О В С К

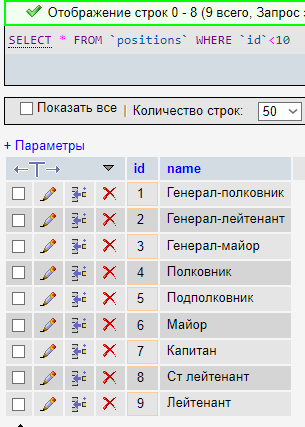
2022г.

**«Диспетчерская МЧС»**

**Запросы с условием на числовые данные (>,<,=, between)**

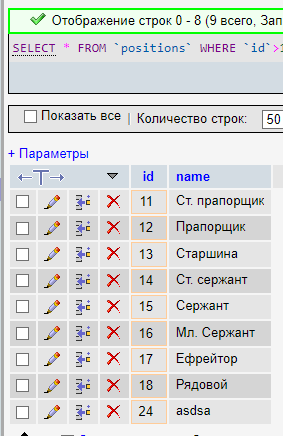
* Вывод строк, у которых id меньше 10

SELECT \* FROM `positions` WHERE `id`<10



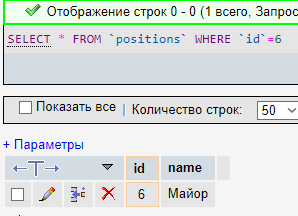
* Вывод строк, у которых id больше 10

SELECT \* FROM `positions` WHERE `id`>10



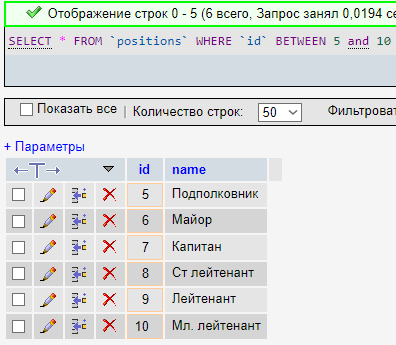
* Вывод строки, у которой id равен 6

SELECT \* FROM `positions` WHERE `id`=6



* Вывод строк, у которых id в диапазоне от 5 до 10

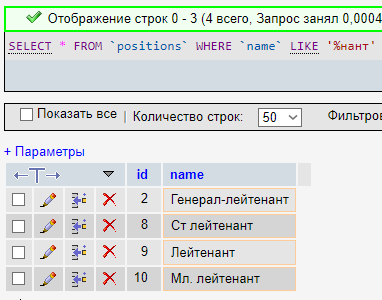
SELECT \* FROM `positions` WHERE `id` BETWEEN 5 and 10



**Запрос с условием на текстовые данные (LIKE, IN);**

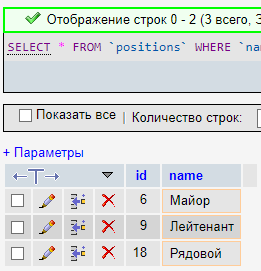
* Вывод строк на основе шаблона «%нант» для поля «name»

SELECT \* FROM `positions` WHERE `name` LIKE '%нант'

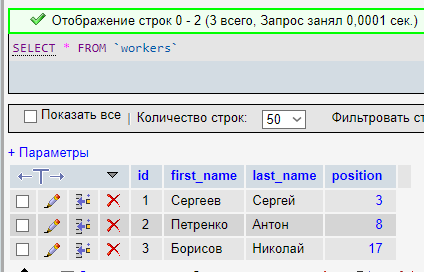


* Вывод строк, где имя столбца равно «Лейтенант», «Майор», «Рядовой»

SELECT \* FROM `positions` WHERE `name` IN ('Лейтенант','Майор','Рядовой')

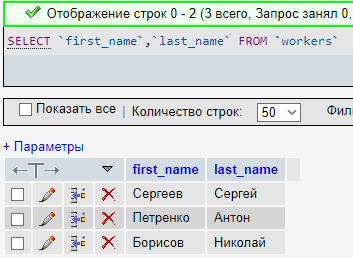


**Запрос с вычисляемым полем;**



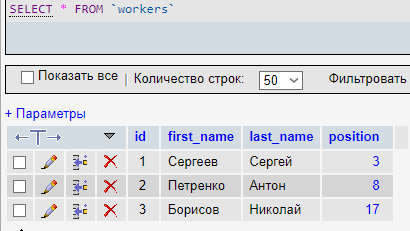
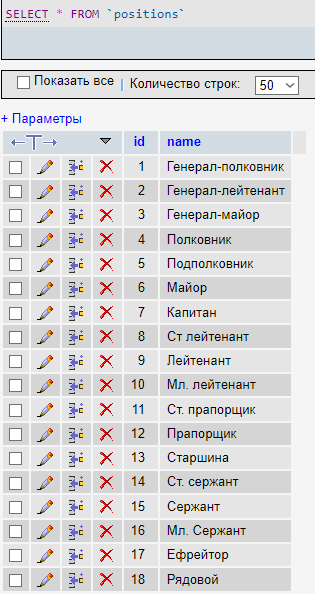
* Вывод строк из таблицы с конкретными столбцами («Фамилия», «Имя»)

SELECT `first\_name`,`last\_name` FROM `workers`



**запрос к нескольким таблицам (без явного указания JOIN);**

* Вывод строк из двух таблиц («Работники», «Должности»), где значение «должности» в таблице «Работники» равна идентификатору в таблице «должности».

SELECT \* FROM `workers`, `positions` WHERE `workers`.`position`=`positions`.`id`

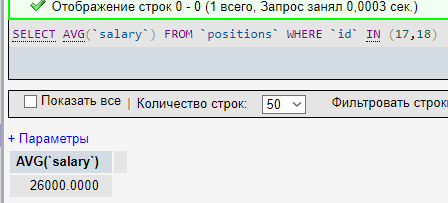


**Запрос с агрегирующей функцией (AVG, SUM, COUNT, MIN, MAX);**

* Выводит среднее значение столбца «Зарплата» у строк, в которых id равен 17,18



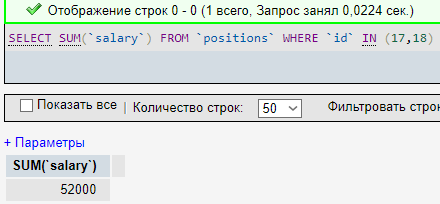
SELECT AVG(`salary`) FROM `positions` WHERE `id` IN (17,18)



* Выводит сумму значений столбца «Зарплата» у строк, в которых id равен 17,18



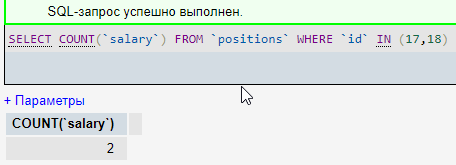
SELECT SUM(`salary`) FROM `positions` WHERE `id` IN (17,18)



* Выводит кол-во строк результата запроса



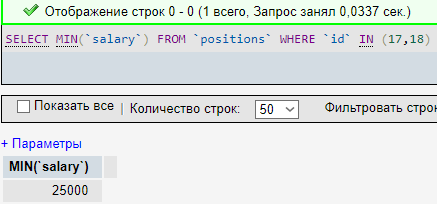
SELECT COUNT(`salary`) FROM `positions` WHERE `id` IN (17,18)



* Выводит МИНимальное значение столбца «Зарплата»



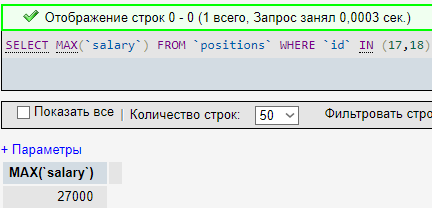
SELECT MIN(`salary`) FROM `positions` WHERE `id` IN (17,18)



* Выводит МАКСимальное значение столбца «Зарплата»



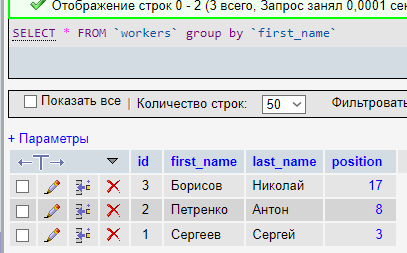
SELECT MAX(`salary`) FROM `positions` WHERE `id` IN (17,18)



**Запрос с группировкой (GROUP BY);**

* Выводит результат с группировкой по столбцу

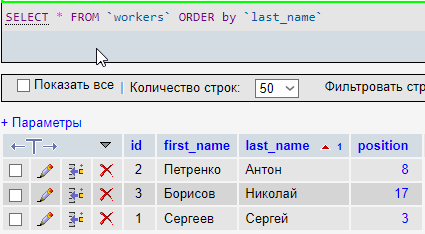
SELECT \* FROM `workers` GROUP by `first\_name`



**Запрос с сортировкой (ORDER BY);**

* Вывод результата с сортировкой по возрастанию

SELECT \* FROM `workers` ORDER by `last\_name`



**Запрос с вложенным подзапросом**

1. SELECT \* FROM `workers`, `positions` WHERE `workers`.`position`=`positions`.`id`



1. SELECT

`workers`.`first\_name`,

`workers`.`last\_name`,

`positions`.`name`,

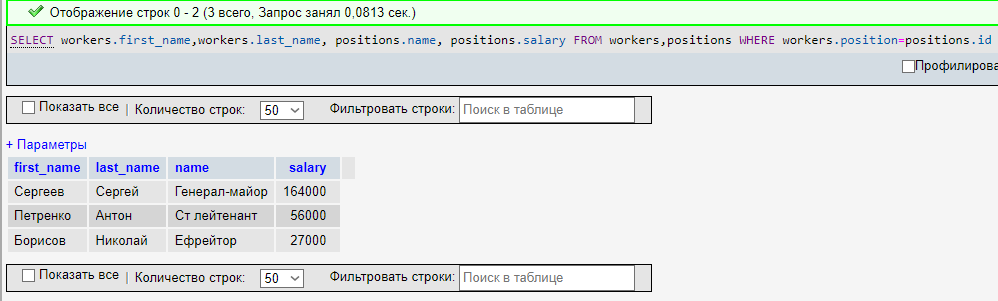
`positions`.`salary`

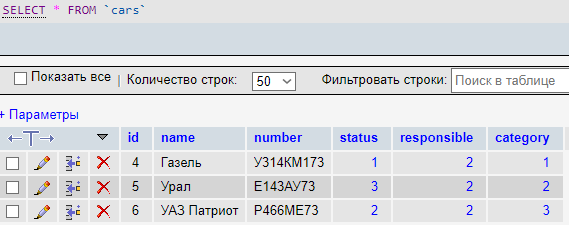
FROM

`workers`,`positions`

WHERE

`workers`.`position`=`positions`.`id`





1. SELECT

`cars`.`name`,

`cars`.`number`,

`statuses`.`name` AS 'Status',

`workers`.`first\_name`,

`workers`.`last\_name`,

`categories`.`name` AS 'category'

FROM

`cars`,

`statuses`,

`workers`,

`categories`

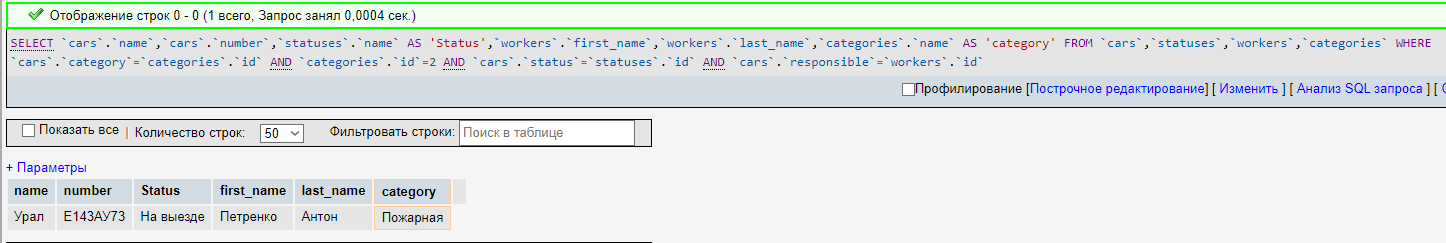
WHERE

`cars`.`category`=`categories`.`id` AND

`categories`.`id`=2 AND

`cars`.`status`=`statuses`.`id` AND

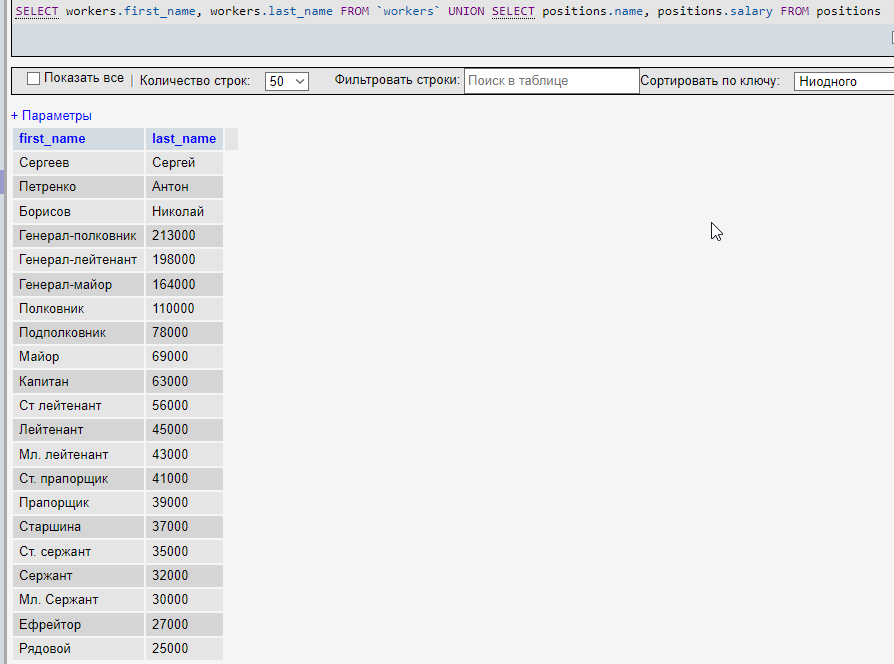
`cars`.`responsible`=`workers`.`id`



**Запрос с оператором UNION**

Объединяет результат запросов и удаляет повторяющиеся строки. Кол-во полей в запросах должно быть одинаковое.

SELECT `workers`.`first\_name`, `workers`.`last\_name` FROM `workers` UNION SELECT `positions`.`name`, `positions`.`salary` FROM `positions`



**Запрос с выражением CASE**

Запрос с условием

SELECT

`id`,

`name`,

`number`,

CASE `category`

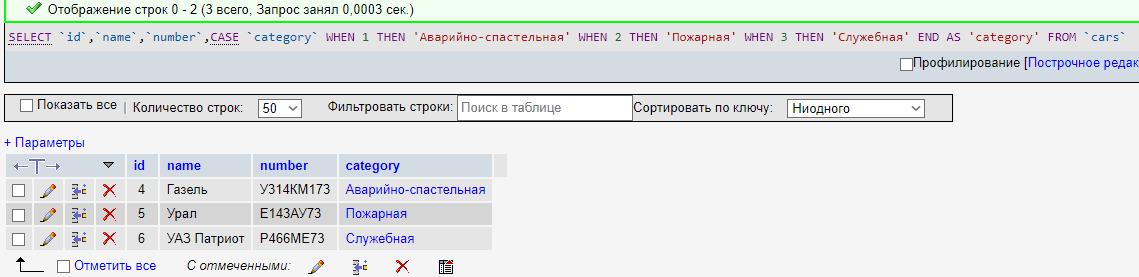
WHEN 1 THEN 'Аварийно-спастельная'

WHEN 2 THEN 'Пожарная'

WHEN 3 THEN 'Служебная'

END AS 'category'

FROM `cars`



**Запрос с оператором JOIN**

1. SELECT

`workers`.`first\_name`,

`workers`.`last\_name`,

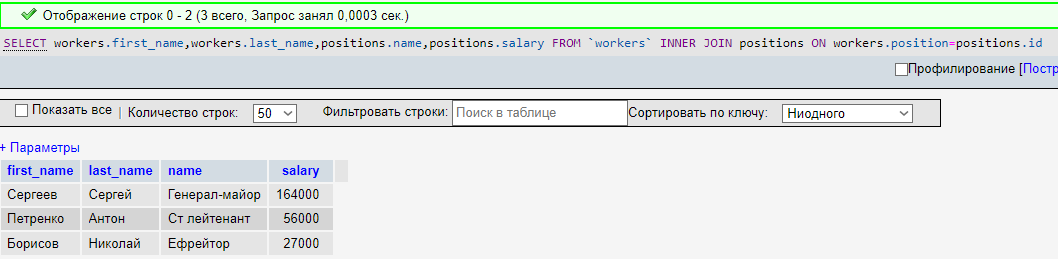
`positions`.`name`,

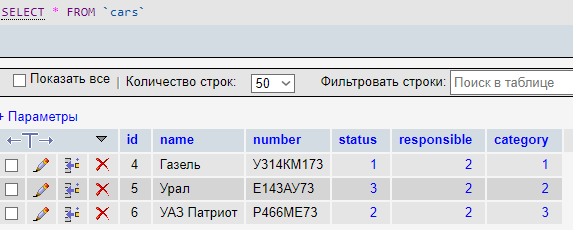
`positions`.`salary`

FROM `workers`

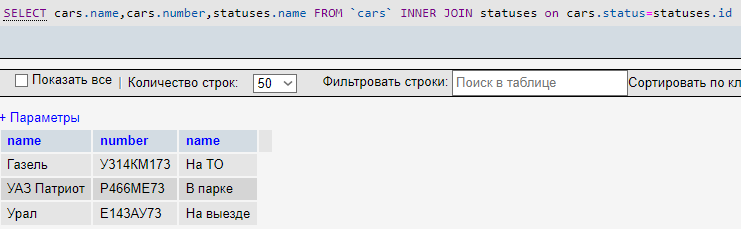
INNER JOIN `positions`

ON `workers`.`position`=`positions`.`id`

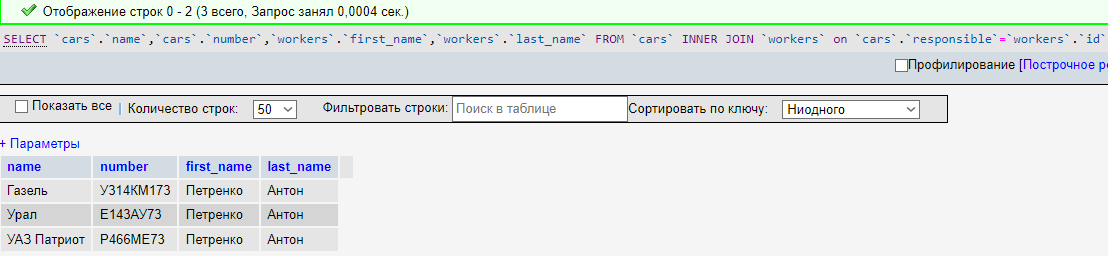




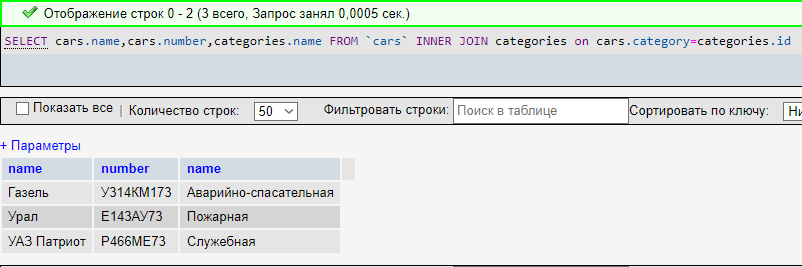
1. SELECT `cars`.`name`,`cars`.`number`,`statuses`.`name` FROM `cars` INNER JOIN `statuses` on `cars`.`status`=`statuses`.`id`



1. SELECT `cars`.`name`,`cars`.`number`,`workers`.`first\_name`,`workers`.`last\_name` FROM `cars` INNER JOIN `workers` on `cars`.`responsible`=`workers`.`id`



1. SELECT `cars`.`name`,`cars`.`number`,`categories`.`name` FROM `cars` INNER JOIN `categories` on `cars`.`category`=`categories`.`id`

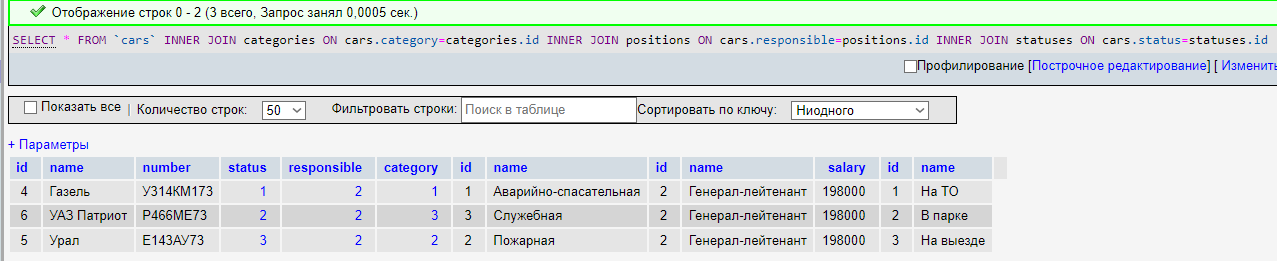


1. SELECT \* FROM `cars`

INNER JOIN `categories` ON `cars`.`category`=`categories`.`id`

INNER JOIN `positions` ON `cars`.`responsible`=`positions`.`id`

INNER JOIN `statuses` ON `cars`.`status`=`statuses`.`id`



**Иерархический запрос**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Машины | | | Контакты | Диспетчерская МЧС |
| Работники (Ответственный) | Категория | Статус |  |  |
| Должности и зарплата |  | |  |  |

SELECT

    `диспетчерская мчс`.`name`,

    `диспетчерская мчс`.`city`,

    `contact`.`address`,

    `contact`.`p\_index`,

    `contact`.`telephone`,

    `contact`.`email`,

    `contact`.`fax`,

    `cars`.`name`,

    `cars`.`number`,

    `statuses`.`name`,

    `workers`.`first\_name`,

    `workers`.`last\_name`,

    `categories`.`name`,

    `positions`.`name`,

    `positions`.`salary`

FROM

    `диспетчерская мчс`,

    `contact`,

    `cars`

INNER JOIN `statuses` ON `cars`.`status`=`statuses`.`id`

INNER JOIN `workers` ON `cars`.`responsible`=`workers`.`id`

INNER JOIN `categories`ON `cars`.`category`=`categories`.`id`

INNER JOIN `positions` ON `workers`.`position`=`positions`.`id`

