

----<https://sqliteonline.com/>

----<https://onecompiler.com/postgresql/3yn4as4qc>

-- 1. Дана схема базы данных в виде следующих отношений. С помощью операторов SQL

-- создать логическую структуру соответствующих таблиц

----DROP TABLE employees;

----DROP TABLE jobs;

----DROP TABLE professions;

----DROP TABLE works;

```
CREATE TABLE employees (  
    employee_id int NOT NULL UNIQUE,  
    last_name varchar(255),  
    address varchar(255),  
    tax int,  
    check (tax >= 0 and tax <= 100),  
    PRIMARY KEY(employee_id)  
);
```

```
CREATE TABLE jobs (  
    job_id int NOT NULL UNIQUE,  
    job_title varchar(255) NOT NULL, --не может быть организации без названия  
    address varchar(255) NOT NULL, --и без адреса тоже  
    tax int,  
    check (tax >= 0 and tax <= 100),  
    PRIMARY KEY(job_id)  
);
```





```
CREATE TABLE professions (  
    profession_id int NOT NULL UNIQUE,  
    profession_title varchar(255),  
    salary_per_hour int,
```

```

        max_hours int,
        PRIMARY KEY(profession_id)
    );

CREATE TABLE works (
    work_number int NOT NULL UNIQUE,
    work_month varchar(255),
    employee_id int,
    job_id int,
    profession_id int,
    hours int,
    salary int,
    PRIMARY KEY(work_number)
);

```

Table		
	employees	<
	jobs	<
	professions	<
	works	<

-- 2. Ввести в ранее созданные таблицы конкретные данные (см. прил. 1).
Использовать

-- скрипт-файл из операторов INSERT или вспомогательную утилиту .

```

insert into employees VALUES(1, 'Пивоваров', 'Канавинский', 10);
insert into employees VALUES(2, 'Махалина', 'Сормовский', 10);
insert into employees VALUES(3, 'Щанников', 'Нижегородский', 15);
insert into employees VALUES(4, 'Воробьев', 'Советский', 15);
insert into employees VALUES(5, 'Александров', 'Советский', 10);

```

```

INSERT into jobs VALUES(1, 'Университет', 'Приокский', 20);

```

```
INSERT into jobs VALUES(2, 'Институт механики', 'Приокский', 10);
INSERT into jobs VALUES(3, 'Технический Университет', 'Нижегородский', 20);
INSERT into jobs VALUES(4, 'НИИ ПМК', 'Нижегородский', 10);
INSERT into jobs VALUES(5, 'Сельхоз. академия', 'Приокский', 20);
INSERT into jobs VALUES(6, 'Академия бизнеса', 'Сормовский', 25);
```

```
INSERT into professions VALUES(1, 'Ассистент', 10000, 40);
INSERT into professions VALUES(2, 'Старший преподаватель', 15000, 35);
INSERT into professions VALUES(3, 'Доцент', 20000, 20);
INSERT into professions VALUES(4, 'Профессор', 25000, 10);
INSERT into professions VALUES(5, 'Мл. научный сотрудник', 7000, 60);
INSERT into professions VALUES(6, 'Ст. научный сотрудник', 10000, 50);
INSERT into professions VALUES(7, 'Зав. лабораторией', 13000, 40);
```

```
INSERT INTO works VALUES(20000, 'Январь', 003, 004, 007, 10, 130000);
INSERT INTO works VALUES(20001, 'Январь', 005, 006, 004, 5, 125000);
INSERT INTO works VALUES(20002, 'Февраль', 001, 006, 001, 35, 350000);
INSERT INTO works VALUES(20003, 'Февраль', 002, 002, 005, 10, 70000);
INSERT INTO works VALUES(20004, 'Февраль', 002, 001, 002, 30, 450000);
INSERT INTO works VALUES(20005, 'Февраль', 002, 005, 001, 10, 100000);
INSERT INTO works VALUES(20006, 'Февраль', 003, 002, 003, 15, 300000);
INSERT INTO works VALUES(20007, 'Апрель', 001, 001, 002, 20, 300000);
INSERT INTO works VALUES(20008, 'Апрель', 002, 005, 006, 40, 400000);
INSERT INTO works VALUES(20009, 'Апрель', 004, 005, 001, 10, 100000);
INSERT INTO works VALUES(20010, 'Май', 002, 002, 002, 20, 300000);
INSERT INTO works VALUES(20011, 'Июнь', 003, 006, 003, 11, 220000);
INSERT INTO works VALUES(20012, 'Июль', 001, 003, 002, 10, 150000);
INSERT INTO works VALUES(20013, 'Июль', 002, 002, 003, 15, 300000);
INSERT INTO works VALUES(20014, 'Август', 004, 002, 004, 8, 200000);
INSERT INTO works VALUES(20015, 'Август', 005, 002, 007, 10, 130000);
INSERT INTO works VALUES(20016, 'Август', 001, 003, 002, 20, 300000);
```

--3. Используя оператор SELECT создать запрос для вывода всех строк каждой таблицы.

--Проверить правильность ввода. При необходимости произвести коррекцию значений

--операторами INSERT, UPDATE, DELETE.

--SELECT * FROM employees

employee_id	last_name	address	tax
1	Пивоваров	Канавинский	10
2	Махалина	Сормовский	10
3	Щанников	Нижегородский	15
4	Воробьев	Советский	15
5	Александров	Советский	10

(5 rows)

--SELECT * FROM jobs

job_id	job_title	address	tax
1	Университет	Приокский	20
2	Институт механики	Приокский	10
3	Технический Университет	Нижегородский	20
4	НИИ ПМК	Нижегородский	10
5	Сельхоз. академия	Приокский	20
6	Академия бизнеса	Сормовский	25

(6 rows)

--SELECT * FROM professions

profession_id	profession_title	salary_per_hour	max_hours
1	Ассистент	10000	40
2	Старший преподаватель	15000	35
3	Доцент	20000	20
4	Профессор	25000	10
5	Мл. научный сотрудник	7000	60
6	Ст. научный сотрудник	10000	50
7	Зав. лабораторией	13000	40

(7 rows)

--SELECT * FROM works

work_number	work_month	employee_id	job_id	profession_id	hours	salary
20000	Январь	3	4	7	10	13000
20001	Январь	5	6	4	5	12500
20002	Февраль	1	6	1	35	35000
20003	Февраль	2	2	5	10	7000
20004	Февраль	2	1	2	30	45000
20005	Февраль	2	5	1	10	10000
20006	Февраль	3	2	3	15	30000
20007	Апрель	1	1	2	20	30000
20008	Апрель	2	5	6	40	40000
20009	Апрель	4	5	1	10	10000
20010	Май	2	2	2	20	30000
20011	Июнь	3	6	3	11	22000
20012	Июль	1	3	2	10	15000
20013	Июль	2	2	3	15	30000
20014	Август	4	2	4	8	20000
20015	Август	5	2	7	10	13000
20016	Август	1	3	2	20	30000

17 rows)

--4. Создать запросы для вывода:

--с) всех различных размеров налогов;

```
--SELECT DISTINCT tax FROM employees
```

```
tax
```

```
-----
```

```
10
```

```
15
```

```
(2 rows)
```

--d) всех различных мест работы;

```
--SELECT job_id, job_title FROM jobs --они все различные
```

job_id	job_title
1	Университет
2	Институт механики
3	Технический Университет
4	НИИ ПМК
5	Сельхоз. академия
6	Академия бизнеса

(6 rows)

--е) всех различных районов проживания сотрудников.

```
--SELECT DISTINCT address FROM employees
```

address
Сормовский
Советский
Канавинский
Нижегородский

(4 rows)

--5. Создав запрос получить следующую информацию:

--с) фамилии и адреса сотрудников, имеющих налог более 8%;

```
--SELECT last_name, address FROM employees WHERE tax > 8
```

last_name	address
Пивоваров	Канавинский
Махалина	Сормовский
Щанников	Нижегородский
Воробьев	Советский
Александров	Советский

(5 rows)

--d) почасовую оплату и название для должностей, в названии которых встречаются

--слова “научный сотрудник”

--SELECT salary_per_hour, profession_title FROM professions

--WHERE profession_title like '%научный сотрудник%'

salary_per_hour	profession_title
7000	Мл. научный сотрудник
10000	Ст. научный сотрудник

(2 rows)

--е) номер, дату и количество часов для тех записей о работе, где плата превышала 100000руб.

--SELECT work_number, work_month, hours FROM works WHERE salary > 100000

work_number	work_month	hours
20000	Январь	10
20001	Январь	5
20002	Февраль	35
20004	Февраль	30
20006	Февраль	15
20007	Апрель	20
20008	Апрель	40
20010	Май	20
20011	Июнь	11
20012	Июль	10
20013	Июль	15
20014	Август	8
20015	Август	10
20016	Август	20

(14 rows)

--6. На основании данных о работе вывести все данные в таком формате:

--с) номер, фамилия сотрудника, дата, количество часов. Отсортировать по количеству часов;

```
--SELECT work_number, last_name, work_month, hours
--FROM employees JOIN works on employees.employee_id = works.employee_id
--ORDER By hours
```

work_number	last_name	work_month	hours
20001	Александров	Январь	5
20014	Воробьев	Август	8
20003	Махалина	Февраль	10
20015	Александров	Август	10
20005	Махалина	Февраль	10
20000	Щанников	Январь	10
20012	Пивоваров	Июль	10
20009	Воробьев	Апрель	10
20011	Щанников	Июнь	11
20013	Махалина	Июль	15
20006	Щанников	Февраль	15
20016	Пивоваров	Август	20
20007	Пивоваров	Апрель	20
20010	Махалина	Май	20
20004	Махалина	Февраль	30
20002	Пивоваров	Февраль	35
20008	Махалина	Апрель	40

(17 rows)

--d) название работы, должность, дата, плата.

```
--SELECT job_title, profession_title, work_month, salary
--FROM (jobs join works on jobs.job_id = works.job_id) JOIN professions On
works.profession_id = professions.profession_id
--Если под "должность" понималось только id, то последний join не нужен
```


job_title	profession_title	work_month	salary
Сельхоз. академия	Ассистент	Апрель	100000
Сельхоз. академия	Ассистент	Февраль	100000
Академия бизнеса	Ассистент	Февраль	350000
Технический Университет	Старший преподаватель	Август	300000
Технический Университет	Старший преподаватель	Июль	150000
Институт механики	Старший преподаватель	Май	300000
Университет	Старший преподаватель	Апрель	300000
Университет	Старший преподаватель	Февраль	450000
Институт механики	Доцент	Июль	300000
Академия бизнеса	Доцент	Июнь	220000
Институт механики	Доцент	Февраль	300000
Институт механики	Профессор	Август	200000
Академия бизнеса	Профессор	Январь	125000
Институт механики	Мл. научный сотрудник	Февраль	70000
Сельхоз. академия	Ст. научный сотрудник	Апрель	400000
Институт механики	Зав. лабораторией	Август	130000
НИИ ПМК	Зав. лабораторией	Январь	130000

(17 rows)

--7. Вывести:

--с) названия организаций, где работали доценты или служащие того же района;

```
--SELECT DISTINCT job_title
--FROM jobs, professions, works, employees
--WHERE works.job_id = jobs.job_id and works.employee_id =
employees.employee_id AND works.profession_id = professions.profession_id
-- and (profession_title = 'Доцент' or jobs.address = employees.address)

job_title
-----
Академия бизнеса
Институт механики
НИИ ПМК
(3 rows)
```

--d) фамилии и размер налога для тех работников, которые имели работу с почасовой

--оплатой менее 15000руб. не ранее января;

--SELECT last_name, employees.tax

--FROM works, employees, professions

--WHERE works.employee_id = employees.employee_id and works.profession_id = professions.profession_id

-- AND salary_per_hour < 15000 and work_month <= 'Январь'

--GROUP by employees.employee_id

last_name	tax
Пивоваров	10
Махалина	10
Щанников	15
Воробьев	15
Александров	10

(5 rows)

--е) название и размер отчислений для организаций, где работал Александров более одного раза.

--SELECT job_title, jobs.tax

--FROM works, jobs, employees

--WHERE works.employee_id = employees.employee_id and works.job_id = jobs.job_id

-- and last_name = 'Александров'

-- GROUP BY job_title, jobs.tax

--HAVING COUNT(last_name) > 1

job_title	tax
-----------	-----

(0 rows)

--f) номер работы, название организации, где работали работники из Советского района.

--Добавить в вывод фамилии таких работников и отсортировать по названию

--организации.

--SELECT work_number, job_title, last_name

--FROM works, jobs, employees

--WHERE works.employee_id = employees.employee_id and works.job_id = jobs.job_id

-- AND employees.address = 'Советский'

--ORDER by job_title

work_number	job_title	last_name
20001	Академия бизнеса	Александров
20014	Институт механики	Воробьев
20015	Институт механики	Александров
20009	Сельхоз. академия	Воробьев

(4 rows)

--8. Создать запрос для модификации всех значений столбца с суммарной величиной платы,

--чтобы он содержал истинную сумму, полученную работником (за вычетом налога).

--UPDATE works

-- set salary =

-- p.salary_per_hour * w.hours * (1 - e.tax * 0.01)

-- FROM (works w JOIN professions p on w.profession_id = p.profession_id)

-- JOIN employees e ON w.employee_id = e.employee_id

-- WHERE works.work_number = w.work_number;

--SELECT * FROM works

k_number	work_month	employee_id	job_id	profession_id	hours	salary
20000	Январь	3	4	7	10	110500
20001	Январь	5	6	4	5	112500
20002	Февраль	1	6	1	35	315000
20003	Февраль	2	2	5	10	63000
20004	Февраль	2	1	2	30	405000
20005	Февраль	2	5	1	10	90000
20006	Февраль	3	2	3	15	255000
20007	Апрель	1	1	2	20	270000
20008	Апрель	2	5	6	40	360000
20009	Апрель	4	5	1	10	85000
20010	Май	2	2	2	20	270000
20011	Июнь	3	6	3	11	187000
20012	Июль	1	3	2	10	135000
20013	Июль	2	2	3	15	270000
20014	Август	4	2	4	8	170000
20015	Август	5	2	7	10	117000
20016	Август	1	3	2	20	270000

rows)

--Еще вариант

--UPDATE works

-- set salary =

-- salary * (1 - (tax * 0.01))

-- FROM employees

-- WHERE employees.employee_id = works.employee_id;

--9. Расширить таблицу с данными о работах столбцом, содержащим величину отчислений

--в пенсионный фонд. Создать запрос для ввода конкретных значений во все строки

--таблицы.

--ALTER TABLE works

-- ADD COLUMN job_tax int;

----не работает--ALTER TABLE works

----не работает-- ADD CHECK (job_tax >= 0 and job_tax <= 100);

```
--UPDATE works
--    set job_tax =
--        round(salary * (1 / (1 - jobs.tax * 0.01) - 1), 0)
--    FROM jobs
--    WHERE works.job_id = jobs.job_id;
--
--SELECT * FROM works
```

work_month	employee_id	job_id	profession_id	hours	salary	job_tax
Январь	3	4	7	10	130000	14444
Январь	5	6	4	5	125000	41667
Февраль	1	6	1	35	350000	116667
Февраль	2	2	5	10	70000	7778
Февраль	2	1	2	30	450000	112500
Февраль	2	5	1	10	100000	25000
Февраль	3	2	3	15	300000	33333
Апрель	1	1	2	20	300000	75000
Апрель	2	5	6	40	400000	100000
Апрель	4	5	1	10	100000	25000
Май	2	2	2	20	300000	33333
Июнь	3	6	3	11	220000	73333
Июль	1	3	2	10	150000	37500
Июль	2	2	3	15	300000	33333
Август	4	2	4	8	200000	22222
Август	5	2	7	10	130000	14444
Август	1	3	2	20	300000	75000

--Вся таблица не влезла

--10. Используя операцию IN (NOT IN) реализовать следующие запросы:

--с) найти такие места работы, где не трудились сотрудники из Приокского района;

```
--SELECT DISTINCT jobs.job_id, jobs.job_title FROM
--jobs, employees, works
--WHERE works.job_id = jobs.job_id and works.employee_id =
employees.employee_id
--And employees.address not in ('Приокский')
--ORDER by jobs.job_id
```

```

-----
job_id |      job_title
-----+-----
      1 | Университет
      2 | Институт механики
      3 | Технический Университет
      4 | НИИ ПМК
      5 | Сельхоз. академия
      6 | Академия бизнеса
(6 rows)

```

--d) запросы задания 7.a и 7.d;

----7с) названия организаций, где работали доценты или служащие того же района;

```

--SELECT DISTINCT job_title
--FROM jobs, professions, works, employees
--WHERE works.job_id = jobs.job_id and works.employee_id =
employees.employee_id AND works.profession_id = professions.profession_id
-- and (profession_title IN ('Доцент') or jobs.address IN
(employees.address))

```

```

      job_title
-----
Академия бизнеса
Институт механики
НИИ ПМК
(3 rows)

```

----7d) фамилии и размер налога для тех работников, которые имели работу с почасовой

----оплатой менее 15000руб. не ранее января;

```

--SELECT last_name, employees.tax
--FROM works, employees, professions
--WHERE works.employee_id = employees.employee_id and works.profession_id =
professions.profession_id

```

```
-- AND salary_per_hour < 15000 and work_month IN ('Январь', 'Февраль',
'Mарт', 'Апрель', 'Май', 'Июнь',
-- 'Июль', 'Август',
'Sентябрь', 'Октябрь', 'Ноябрь', 'Декабрь')
--GROUP by employees.employee_id
```

```
last_name | tax
-----+-----
Пивоваров | 10
Махалина  | 10
Щанников  | 15
Воробьев  | 15
Александров | 10
(5 rows)
```

--е) определить должности, на которых работал Пивоваров более раза.

```
--SELECT profession_title
--FROM works, professions, employees
--WHERE works.profession_id = professions.profession_id AND works.employee_id
= employees.employee_id
-- AND employees.last_name IN ('Пивоваров')
--GROUP BY employees.employee_id, professions.profession_title
--HAVING COUNT(works.work_number) > 1
-----
profession_title
-----
Старший преподаватель
(1 row)
```

--11. Используя операции ALL-ANY реализовать следующие запросы:

--с) найти должность с самой высокой почасовой оплатой;

```
--SELECT DISTINCT profession_title
--FROM professions
--WHERE professions.salary_per_hour >= ALL
```

```
-- (SELECT p.salary_per_hour FROM professions p)
profession_title
-----
Профессор
(1 row)
```

--d) найти место работы, где на самой низкооплачиваемой должности работал Воробьев;

```
--SELECT jobs.job_id, jobs.job_title
--FROM works, jobs, professions, employees
--WHERE works.profession_id = professions.profession_id AND works.employee_id
= employees.employee_id AND works.job_id = jobs.job_id
-- AND last_name = 'Воробьев'
-- AND professions.profession_id = ANY
-- (
--     SELECT help_table.profession_id
--     FROM
--         (
--             SELECT DISTINCT j.job_title, p.profession_title,
--             p.profession_id,
--             min(p.salary_per_hour) over(partition by j.job_id) as
min_sal
--             FROM (jobs j join works w on j.job_id = w.job_id) join
professions p on w.profession_id = p.profession_id
--             ) as help_table
--     )
-- )
job_id | job_title
-----+-----
      5 | Сельхоз. академия
      2 | Институт механики
(2 rows)
```

--e) среди работавших в университете найти сотрудника с самым низким налогом;


```
--SELECT employees.employee_id, employees.last_name
--FROM works, jobs, professions, employees
--WHERE works.profession_id = professions.profession_id AND works.employee_id
= employees.employee_id AND works.job_id = jobs.job_id
--    AND jobs.job_id = 1
--    AND employees.tax <= ALL
--        (
--            SELECT e.tax FROM works w, jobs j, professions p, employees e
--                WHERE w.profession_id = p.profession_id
--                    AND w.employee_id = e.employee_id
--                    AND w.job_id = j.job_id
--                    AND j.job_id = 1
--        )
----LIMIT 1 --если нужен строго 1 сотрудник
```

```
employee_id | last_name
-----+-----
          2 | Махалина
          1 | Пивоваров
(2 rows)
```

--f) запрос задания 7.с.

----7с) названия организаций, где работали доценты или служащие того же района;

```
-- SELECT DISTINCT job_title
-- FROM jobs, professions, works, employees
-- WHERE works.job_id = jobs.job_id AND works.employee_id =
employees.employee_id AND works.profession_id = professions.profession_id
--    AND (
--        professions.profession_id = ALL
--        (
--            SELECT p.profession_id
--            FROM professions p
--            WHERE p.profession_title = 'Доцент'
```

```
--      )
--      OR jobs.address = employees.address
--      )
```

job_title

Академия бизнеса
Институт механики
НИИ ПМК

(3 rows)

-- 12. Используя операцию UNION получить адреса проживания сотрудников и места

-- расположения организаций.

```
-- SELECT employees.address FROM employees
```

```
-- UNION
```

```
-- SELECT jobs.address FROM jobs
```

address

Приокский
Канавинский
Нижегородский
Сормовский
Советский

(5 rows)

-- 13. Используя операцию EXISTS (NOT EXISTS) реализовать нижеследующие запросы. В

-- случае, если для текущего состояния БД запрос будет выдавать пустое множество

-- строк, требуется указать какие добавления в БД необходимо провести.

-- с) найти сотрудников, работавших на должности доцента во всех организациях;

```
-- SELECT employee_id, last_name
-- FROM employees
-- WHERE EXISTS
-- (
--     SELECT e.employee_id
--     FROM employees e, works w, professions p, jobs j
--     WHERE w.employee_id = e.employee_id AND w.profession_id =
p.profession_id AND w.job_id = j.job_id
--     AND employees.employee_id = e.employee_id
--     GROUP BY e.employee_id
--     HAVING COUNT(j.job_id) = 6
-- )
```

```
employee_id | last_name
-----+-----
          2 | Махалина
(1 row)
```

-- d) найти места работы на которых работали все сотрудники из Приокского или
-- Сормовского районов;

```
-- SELECT DISTINCT jobs.job_id, jobs.job_title
-- FROM works, jobs, employees
-- WHERE works.job_id = jobs.job_id AND works.employee_id =
employees.employee_id
-- AND NOT EXISTS
-- (
--     SELECT e.employee_id
--     FROM employees e
--     WHERE e.address NOT IN ('Приокский', 'Сормовский') AND e.employee_id =
employees.employee_id
-- )
```

```

-----+-----
job_id |      job_title
-----+-----
      1 | Университет
      2 | Институт механики
      5 | Сельхоз. академия
(3 rows)

```

-- е) определить должности на которых не работали сотрудники в организациях чужих

-- районов;

```

-- SELECT DISTINCT professions.profession_id, professions.profession_title
-- FROM works, jobs, employees, professions
-- WHERE works.job_id = jobs.job_id AND works.employee_id =
employees.employee_id
--     AND works.profession_id = professions.profession_id
--     AND NOT EXISTS
--     (
--         SELECT e.employee_id
--         FROM employees e
--         WHERE e.address != jobs.address AND e.employee_id =
employees.employee_id
--     )

```

```

INSERT INTO
profession_id | profession_title
-----+-----
              7 | Зав. лабораторией
(1 row)

```

-- f) найти сотрудника и должность, в которой этот сотрудник проработал во всех

-- организациях Нижегородского или Сормовского районов

```

-- SELECT DISTINCT employees.employee_id, employees.last_name,
--     professions.profession_id, professions.profession_title

```

```

-- FROM works, jobs, employees, professions
-- WHERE works.job_id = jobs.job_id AND works.employee_id =
employees.employee_id
-- AND works.profession_id = professions.profession_id
-- AND
-- (
--     EXISTS
--     (
--         SELECT w.employee_id
--         FROM works w
--         WHERE w.work_number = works.work_number AND jobs.address =
'Нижегородский'
--         GROUP BY w.employee_id
--         HAVING COUNT(DISTINCT w.job_id) = 1
--     )
-- OR
-- EXISTS
-- (
--     SELECT w.employee_id
--     FROM works w
--     WHERE w.work_number = works.work_number AND jobs.address =
'Сормовский'
--     GROUP BY w.employee_id
--     HAVING COUNT(DISTINCT w.job_id) = 1
-- )
-- )

```

employee_id	last_name	profession_id	profession_title
1	Пивоваров	1	Ассистент
1	Пивоваров	2	Старший преподаватель
3	Щанников	3	Доцент
3	Щанников	7	Зав. лабораторией
5	Александров	4	Профессор

(5 rows)

-- 14. Реализовать запросы с использованием агрегатных функций:
-- с) найти число различных работников, трудившихся в должности ассистента до
-- ноября;

```
-- SELECT COUNT(employees.employee_id)
-- FROM works, employees, professions
-- WHERE works.employee_id = employees.employee_id AND works.profession_id =
professions.profession_id
--   AND professions.profession_title like '%ссистент%'
--   AND works.work_month NOT IN ('Ноябрь', 'Декабрь')

count
-----
      3
(1 row)
```

-- d) определить среднюю величину отчислений для тех организаций, где
трудились

-- сотрудники с налогом менее 15%;

```
-- ALTER TABLE works
--   ADD COLUMN job_tax int;

-- UPDATE works
--   set job_tax =
--       round(salary * (1 / (1 - jobs.tax * 0.01) - 1), 0)
--       FROM jobs
--       WHERE works.job_id = jobs.job_id;

-- SELECT DISTINCT works.job_id,
--   ROUND(AVG(works.job_tax) OVER(PARTITION BY works.job_id), 0) as avg_tax
-- FROM works, employees
-- WHERE works.employee_id = employees.employee_id
--   AND employees.tax < 15
-- ORDER BY works.job_id;
```

```
-- ALTER TABLE works
-- DROP COLUMN job_tax;
```

```
job_id | avg_tax
-----+-----
      1 |   93750
      2 |   22222
      3 |   56250
      5 |   62500
      6 |   79167
(5 rows)
```

-- е) найти число сотрудников, работавших до марта в университете;

```
-- SELECT COUNT(DISTINCT works.employee_id)
-- FROM works JOIN jobs ON works.job_id = jobs.job_id
-- WHERE jobs.job_title = 'Университет'
-- AND works.work_month IN ('Январь', 'Февраль')
```

```
count
-----
      1
(1 row)
```

-- f) какие работники получали плату выше средней.

```
-- SELECT DISTINCT employees.employee_id, employees.last_name
-- FROM works JOIN employees ON works.employee_id = employees.employee_id
-- WHERE works.salary >
-- (SELECT AVG(w.salary) FROM works w)
-- ORDER BY employees.employee_id
```

```

employee_id | last_name
-----+-----
          1 | Пивоваров
          2 | Махалина
          3 | Щанников
(3 rows)

```

-- 15. Используя средства группировки реализовать следующие запросы:

-- с) найти должности, на которых работало более трех человек;

```

-- SELECT works.profession_id, professions.profession_title
-- FROM works JOIN professions ON works.profession_id =
-- professions.profession_id
-- GROUP BY works.profession_id, professions.profession_title
-- HAVING COUNT(employee_id) > 3

```

```

profession_id | profession_title
-----+-----
          2 | Старший преподаватель
(1 row)

```

-- d) найти для каждого сотрудника максимальную получаемую плату;

```

-- SELECT DISTINCT employees.employee_id, employees.last_name,
--    MAX(works.salary) OVER(PARTITION BY works.employee_id)
-- FROM works JOIN employees ON works.employee_id = employees.employee_id
-- ORDER BY employees.employee_id ASC

```

```

employee_id | last_name | max
-----+-----+-----
          1 | Пивоваров | 350000
          2 | Махалина  | 450000
          3 | Щанников  | 300000
          4 | Воробьев  | 200000
          5 | Александров | 130000
(5 rows)

```


-- е) какие организации за месяц тратили на зарплату более 1000000;

```
-- SELECT works.job_id, jobs.job_title
-- FROM works JOIN jobs ON works.job_id = jobs.job_id
-- GROUP BY works.job_id, jobs.job_title
-- HAVING SUM(works.salary) > 1000000
```

job_id	job_title
2	Институт механики

(1 row)

-- f) получить для каждой организации из Нижегородского или Сормовского района

-- среднее число часов, которое набирали сотрудники.

```
-- SELECT DISTINCT jobs.job_id, jobs.job_title,
--    ROUND(AVG(works.hours) OVER(PARTITION BY works.job_id), 0) as avg_hours
-- FROM works JOIN jobs ON works.job_id = jobs.job_id
-- WHERE jobs.address IN ('Нижегородский', 'Сормовский')
```

job_id	job_title	avg_hours
3	Технический Университет	15
4	НИИ ПМК	10
6	Академия бизнеса	17

(3 rows)