Нумерация и названия пунктов, номера заданий в этом проекте соответствуют названиям юнитов и заданиям в модуле PROJECT-2

2. ПРЕДВАРИТЕЛЬНЫЙ АНАЛИЗ ДАННЫХ

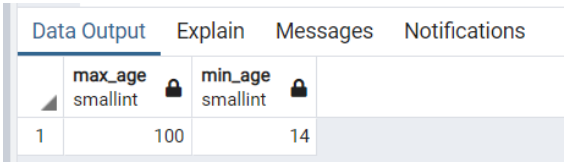
Задание 2.1

Рассчитайте максимальный возраст *(max_age*) кандидата в таблице.

Задание 2.2

Рассчитайте минимальный возраст (*min_age*) кандидата в таблице.

```
SELECT
	-- запрос максимального и минимального возрастов в
	-- столбце age с использованием агрегатных функций
	MAX(c.age) max_age,
	MIN(c.age) min_age
FROM hh.candidate c
```



Data Output | Explain | Messages | Notifications

	max_age smallint	min_age smallint
1	100	14

Ответ:

Максимальный возраст *(max_age)*: 100 лет

Минимальный возраст *(min_age)*: 14 лет

Выводы по заданиям 2.1 и 2.2:

100-ий кандидат - это явно аномальный выброс.

Подлежит очистке.

А вот 14-го кандидата с трудом, но можно допустить, поэтому этот возраст пока оставляем.

Исходя из этого, ещё до конкретного анализа, видно, что таблица "грязная", следует провести её очистку.

****

Задание 2.3

Рассчитать для каждого возраста *(age)* сколько человек *(cnt)* этого возраста у нас есть.

Отсортировать результат по возрасту в обратном порядке.

```
SELECT
	-- запрос возрастов (age) с подсчётом
	-- их количества (функция count())
	c.age,
	COUNT(c.age) cnt
FROM hh.candidate c
GROUP BY age
ORDER BY age DESC
```

Data Output | Explain | Messages | Notifications

	age smallint	cnt bigint
1	100	1
2	77	1
3	76	1
4	73	4
5	72	3
6	71	4
7	70	3
8	69	4
9	68	4
10	67	9

Вывод:

По результатам итоговой таблицы можно сделать вывод о необходимости очистки начиная с 76-его возраста и выше, а также с 16-го (включительно) и ниже.

Возрасты, где количество кандидатов по 1 человеку с “подозрительными” возрастами (14, 76, 77, 100) явно необходимо отфильтровывать.

Для более детального обзора ниже была проведена группировка по возрастам.

****

Задание 2.4

Средний возраст занятых в экономике России составляет 39.7 лет. Округлим это значение до 40. Найти количество кандидатов, которые старше данного возраста.

Отфильтровать "ошибочный" возраст 100.

```
SELECT
      -- общее количество id кандидатов
      COUNT(c.id)
FROM hh.candidate c
-- Фильтрация по условию: больше 40 лет
-- и меньше 100 (не включительно)
WHERE age > 40 AND age < 100
```

Data Output | Explain | Messages | Notifications

	count bigint
1	6263

Ответ:

Количество кандидатов старше 40 лет (исключая 100): 6263

Вывод:

По общему количеству кандидатов число кандидатов от 40 лет и выше имеет невысокий процент.

В цене кандидаты молодые и перспективные.

****

Дополнительно была проведена группировка по возрастам кандидатов.

```
-- с помощью with разбиваем количество кандидатов
-- по возрастным группам: до 20, 20-40, 40-60, 60-80, 80-100
WITH
-- total общее количество кандидатов
total AS (SELECT COUNT(cand.age) AS total_count
            FROM hh.candidate cand),
-- group20 количество кандидатов до 20 лет
group20     AS (SELECT COUNT(cand.age) AS group_count_20
            FROM hh.candidate cand
            WHERE cand.age <= 20),
-- group24 количество кандидатов от 20 до 40 лет
group24     AS (SELECT COUNT(cand.age) AS group_count_24
            FROM hh.candidate cand
            WHERE cand.age > 20 AND cand.age <= 40),
-- далее запросы идентичны, меняется только условие фильтрации
group46     AS (SELECT COUNT(cand.age) AS group_count_46
            FROM hh.candidate cand
            WHERE cand.age > 40 AND cand.age <= 60),
group68     AS (SELECT COUNT(cand.age) AS group_count_68
            FROM hh.candidate cand
            WHERE cand.age > 60 AND cand.age <= 80),
group81     AS (SELECT COUNT(cand.age) AS group_count_81
            FROM hh.candidate cand
            WHERE cand.age > 80 AND cand.age <= 100)
(SELECT
-- проводится запрос из группы до 20 лет
      'до 20 (включительно)' "Возраст (лет)",
      group20.group_count_20 "Количество (чел.)",
      -- вычисление доли от общего количества
-- в процентах, округление до 4 знака
      round(group20.group_count_20 * 100.0 / total.total_count, 4) "Доля в
процентах (%)"
FROM total, group20)
UNION ALL
(SELECT
      -- запрос количества кандидатов из группы от 20 до 40 лет
      'от 20 до 40',
      group24.group_count_24,
      -- вычисление доли от общего количества в процентах,
      -- округление до 4 знака
      round(group24.group_count_24 * 100.0 / total.total_count, 4)
FROM total, group24)
UNION ALL
-- далее запросы идентичны, меняются только названия групп
(SELECT
      'от 40 до 60',
      group46.group_count_46,
      round(group46.group_count_46 * 100.0 / total.total_count, 4)
FROM total, group46)
UNION ALL
(SELECT
      'от 60 до 80',
```

```
	group68.group_count_68,
	round(group68.group_count_68 * 100.0 / total.total_count, 4)
FROM total, group68)
UNION ALL
(SELECT
	'от 80 до 100',
	group81.group_count_81,
	round(group81.group_count_81 * 100.0 / total.total_count, 4)
FROM total, group81)
UNION ALL
(SELECT
	'ВСЕГО',
	total.total_count,
	-- вычисление процента, округление и устранение ненужных нулей
	-- после точки (запятой)
	CAST(CAST(round(total.total_count * 100.00 / total.total_count, 4) AS
decimal(9,6)) AS float)
FROM total);
```

Data Output | Explain | Messages | Notifications

	Возраст (лет) text	Количество (чел.) bigint	Доля в процентах (%) double precision
1	до 20 (включительно)	660	1.4751
2	от 20 до 40	37820	84.5253
3	от 40 до 60	6093	13.6175
4	от 60 до 80	170	0.3799
5	от 80 до 100	1	0.0022
6	ВСЕГО	44744	100

Вывод:

Разбивка по группам возрастов явно выделила лидирующую группу. Это кандидаты в возрасте от 20 до 40 лет. Их доля – 84.5%. Это лидерство можно объяснить несколькими причинами:

1. люди молодые, энергичные, желающие многого добиться, получать хорошую зарплату;
2. работодатель жалует именно таких людей, инициативных и работающих, трудоголиков;
3. спрос на более зрелый возраст гораздо ниже, всего 13.6%, а также, помимо всего, основное количество опытных программистов приходится именно на возраст от 20 до 40 лет. Точно не знаю, проводил ли кто сравнительный анализ по этой теме, но надеюсь, что в будущем ситуация изменится.

“Унылый и одинокий” столетний выброс необходимо удалить.

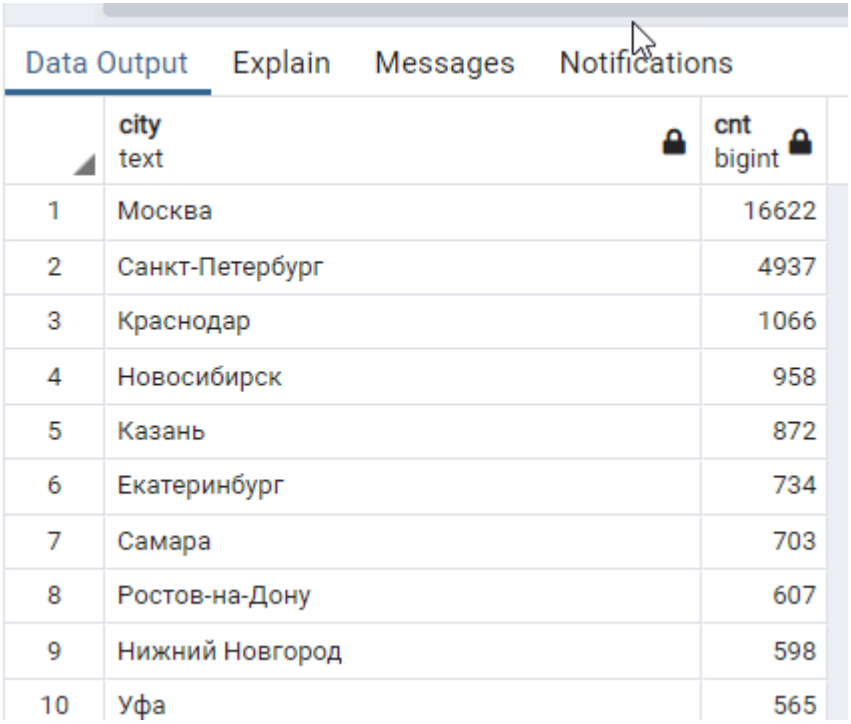
3. ГЛОБАЛЬНЫЙ АНАЛИЗ ПОКАЗАТЕЛЕЙ

Задание 3.1

Необходимо узнать сколько (*cnt*) у нас кандидатов из каждого города (*city*).
Отсортировать результат по количеству в обратном порядке.

```
SELECT
	-- запрос столбца title
	-- запрос количества id кандидатов
```

```
c.title city,
	COUNT(cand.id) cnt
FROM
	hh.candidate cand
	JOIN hh.city c ON cand.city_id = c.id
-- группировка по алиасу city с последующей
-- сортировкой по алиасу cnt
GROUP BY city
ORDER BY cnt DESC
```



Data Output | Explain | Messages | Notifications

	city text	cnt bigint
1	Москва	16622
2	Санкт-Петербург	4937
3	Краснодар	1066
4	Новосибирск	958
5	Казань	872
6	Екатеринбург	734
7	Самара	703
8	Ростов-на-Дону	607
9	Нижний Новгород	598
10	Уфа	565

Вывод:

Лидер по количеству кандидатов – это Москва. Вторым идёт Санкт-Петербург, отставая от Москвы примерно в 3.3 раза. Много кандидатов живут в Москве. Много компаний находится именно в Москве. В Москве самая высокая зарплата по стране. В Москве выше перспективы карьерного роста. Много чего ещё преобладает в Москве, этим и объясняется такой большой отрыв. Лично для меня самый главный вывод - ищи работу в Москве, Санкт-Петербурге, в крупных городах.

Спускаясь дальше по таблице, видно, что чем ниже рейтинг города, чем он малочисленней и дальше от центра, тем меньше имеет кандидатов. Это тоже понятно – кому захочется работать в глухомани, на низкую зарплату, без всяких перспектив?

Задание 3.2

Каких кандидатов из Москвы устроит "*проектная работа*"?

Формат выборки: *gender*, *age*, *desirable_occupation*, *city*, *employment_type*.

Отсортировать по *id* кандидата.

```
SELECT
	-- запрос по столбцам, указанных в задании
	cand.gender,
	cand.age,
	cand.desirable_occupation,
	c.title city,
	cand.employment_type
FROM
```

```
        hh.candidate cand
        -- формирование join с условием вывода наименования города
        -- с именем "Москва" и строк, содержащих в employment type
        -- подстроку "проектная работа"
        JOIN hh.city c ON cand.city_id = c.id AND c.title = 'Москва'
        AND cand.employment_type like '%проектная работа%'
GROUP BY cand.id, gender, age, desirable_occupation, c.title, employment_type
ORDER BY cand.id
```

Data Output | Explain | Messages | Notifications

	gender character (1)	age smallint	desirable_occupation text	city text
1	M	38	Веб-разработчик (HTML / CSS / JS / PHP / базы данных; фреймворки, дизайн, интерфейсы, CMS)	Моск
2	M	31	Специалист	Моск
3	F	42	pre-sale инженер, pre-sale менеджер	Моск
4	M	49	Дежурный администратор	Моск
5	M	29	Главный инженер проекта	Моск
6	M	22	Программист C++	Моск
7	F	29	Технический специалист	Моск
8	M	32	IT Operations Coordinator	Моск
9	M	23	Инженер-связист,системный администратор	Моск

Вывод:

В Москве можно найти *IT*-кандидата практически любого профиля с желаемой работой, которая включает в себя проектные разработки. Большее количество кандидатов ищет работу с "частичной/полной занятостью и проектной работой".

Задание 3.3

Отфильтровать следующие *IT*-профессии — разработчик, аналитик, программист.
Обратите внимание, что данные названия могут быть написаны как с большой, так и с маленькой буквы.
Отсортируйте результат по *id* кандидата.

```
SELECT
        -- запрос по столбцам, указанных в предыдущем задании
        cand.gender,
        cand.age,
        cand.desirable_occupation,
        c.title city,
        cand.employment_type
FROM
        hh.city c
        JOIN hh.candidate cand ON cand.city_id = c.id
WHERE
        -- формирование необходимого условия фильтрации
        c.title = 'Москва'
        AND cand.employment_type ilike '%проектная работа%'
        AND (cand.desirable_occupation ilike '%разработчик%'
        OR cand.desirable_occupation ilike '%аналитик%'
        OR cand.desirable_occupation ilike '%программист%')
ORDER BY cand.id
```

Data Output | Explain | Messages | Notifications

	gender character (1)	age smallint	desirable_occupation text	city text	employment_t text
1	M	38	Веб-разработчик (HTML / CSS / JS / PHP / базы данных; фреймворки, дизайн, интерфейсы, CMS)	Москва	частичная за
2	M	22	Программист C++	Москва	проектная ра
3	M	25	Frontend-разработчик	Москва	стажировка,
4	M	30	Программист	Москва	частичная за
5	M	35	Ruby / Rails разработчик	Москва	частичная за
6	M	28	Программист микроконтроллеров	Москва	стажировка,
7	M	36	Программист-разработчик	Москва	частичная за
8	M	25	Аналитик	Москва	проектная ра
9	M	38	Инженер, программист C/C++, разработчик ПО	Москва	частичная за
10	F	54	Ведущий инженер программист	Москва	частичная за
11	M	41	Программист C++	Москва	проектная ра
12	M	43	Программист C++, мультимедийные приложения	Москва	частичная за

Вывод:

Мы получили общую картину по трём популярным *IT*-профессиям. Количественный сравнительный анализ по ней сделать затруднительно, поэтому **дополнительно** создал новый запрос, который разбивает по группам *IT*-профессии и полу кандидатов, одновременно подсчитывая их количество. Все запросы проводятся для г. Москвы.

```
-- создание CTE prof
WITH prof AS
(SELECT
	-- запрос по соответствующим столбцам
	cand.gender gen,
	cand.age,
	cand.desirable_occupation des,
	cand.employment_type emp
FROM
	hh.city c
	-- условие соединения по id городов
	JOIN hh.candidate cand ON cand.city_id = c.id
WHERE
	-- фильтрация запроса
	c.title = 'Москва' -- согласно условия задачи-г. Москва
	AND cand.employment_type ilike '%проектная работа%'
	AND (cand.desirable_occupation ilike '%разработчик%'
	OR cand.desirable_occupation ilike '%аналитик%'
	OR cand.desirable_occupation ilike '%программист%')
)
(SELECT
	-- запрос по группе профессии "разработчик"
	'Москва' "Город",
	'Разработчик' "IT-профессия",
	count(prof.gen) "Количество (чел.)",
	-- количественный запрос с условием по полу кандидата
	count(CASE WHEN prof.gen = 'M' THEN 'M' END) AS "Мужчины (чел.)",
	count(CASE WHEN prof.gen = 'F' THEN 'F' END) AS "Женщины (чел.)"
FROM prof
WHERE
	-- фильтрация запроса
	prof.emp ilike '%проектная работа%'
	AND prof.des ilike '%разработчик%')
UNION ALL
-- объединение таблицы с последующей
-- остальные запросы идентичны, меняется только группа профессии
```

```
(SELECT
      'Москва',
      'Аналитик',
      count(prof.gen),
      count(CASE WHEN prof.gen = 'M' THEN 'M' END),
      count(CASE WHEN prof.gen = 'F' THEN 'F' END)
FROM prof
WHERE
      prof.emp ilike '%проектная работа%'
      AND prof.des ilike '%аналитик%')
UNION ALL
(SELECT
      'Москва',
      'Программист',
      count(prof.gen),
      count(CASE WHEN prof.gen = 'M' THEN 'M' END),
      count(CASE WHEN prof.gen = 'F' THEN 'F' END)
FROM prof
WHERE
      prof.emp ilike '%проектная работа%'
      AND prof.des ilike '%программист%')
-- сортировка по общему количеству кандидатов
ORDER BY "Количество (чел.)"
```

Data Output | Explain | Messages | Notifications

	Город text	IT-профессия text	Количество (чел.) bigint	Мужчины (чел.) bigint	Женщины (чел.) bigint
1	Москва	Аналитик	137	94	43
2	Москва	Разработчик	284	269	15
3	Москва	Программист	420	385	35

Вывод:

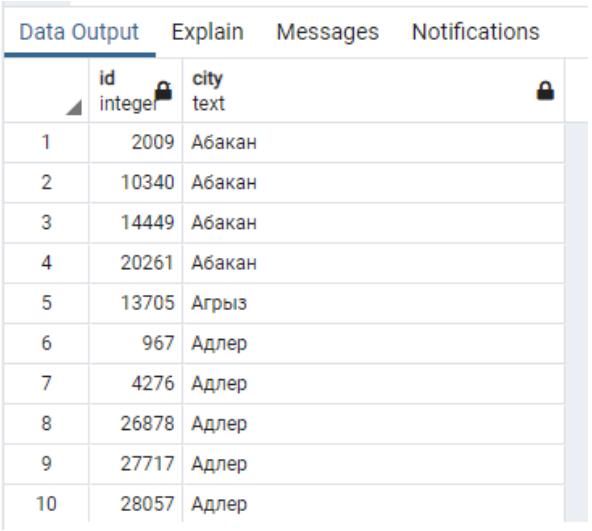
Самая распространённая профессия – программист. 420 кандидатов, хотят сменить её. Преобладают мужчины, женщин очень мало. Если, например, разработчиков-мужчин – 269 человек, то разработчиков-женщин всего 15. Почему так происходит? Можно сделать вывод, что в данном направлении *IT*-индустрии (если не во всей) женщины идут работать крайне неохотно. По каким причинам? Для этого необходимо проводить целое исследование с привлечением различных специалистов: *IT*-экспертов, психологов, необходимо учитывать социальную ситуацию и многое другое, что не входит в задачу этого проекта.

Задание 3.4

Вывести номера и города кандидатов, у которых занимаемая должность совпадает с желаемой. Формат выборки: *id*, *city*.
Отсортировать по городу и *id* кандидата.

```
SELECT
    -- запрос по указанным столбцам
      cand.id id,
      c.title city
FROM
      hh.city c
    -- условие соединения по id городов и
    -- соответствию профессий (нижний регистр)
      JOIN hh.candidate cand ON cand.city_id = c.id
```

```
    AND current_occupation = desirable_occupation
-- сортировка по городу и id
ORDER BY city, id
```



Data Output | Explain | Messages | Notifications

	id integer	city text
1	2009	Абакан
2	10340	Абакан
3	14449	Абакан
4	20261	Абакан
5	13705	Агрыз
6	967	Адлер
7	4276	Адлер
8	26878	Адлер
9	27717	Адлер
10	28057	Адлер

Вывод:

Запрос вывел развёрнутую таблицу по каждому городу и кандидату, у которого занимаемая должность совпадает с желаемой. Таблица большая, но при пролистывании видно, что лидеры здесь – Москва и Санкт-Петербург. **Дополнительно** немного изменю код, чтобы таблица запроса стала более наглядной и отсортирую по *id* в обратном порядке.

```
-- создание CTE total и cnt
WITH total AS
(SELECT
      -- запрос количества всех кандидатов
      count(*) cnt_all
 FROM hh.candidate
),
cnt AS
(SELECT
      -- запрос количества кандидатов с совпадающими должностями
      count(*) cnt_part
 FROM
      hh.city ct
      JOIN hh.candidate can ON can.city_id = ct.id
      AND lower(current_occupation) = lower(desirable_occupation)
)
(SELECT
      -- запрос всех кандидатов, формирование столбцов
      'ВСЕГО КАНДИДАТОВ:' "Город",
      total.cnt_all "Количество",
      -- вычисление доли в процентах, округление и удаление
      -- ненужных нулей после точки (запятой)
      CAST(CAST(round(total.cnt_all   *   100.0   /   total.cnt_all,   3)   AS
decimal(9,6)) AS float) "Доля в процентах (%)"
 FROM total, cnt
)
UNION ALL
(SELECT
      'Количество кандидатов с совпадающими должностями:',
```

```
      cnt.cnt_part,
      CAST(CAST(round(cnt.cnt_part   *   100.0   /   total.cnt_all,   3)   AS
decimal(9,6)) AS float)
 FROM total, cnt
 group by cnt.cnt_part, total.cnt_all
)
UNION ALL
(SELECT
      -- запрос по указанным столбцам с подсчётом
      -- количества кандидатов и вычислением их доли в процентах
      c.title,
      count(*),
      CAST(CAST(round(count(*)  * 100.0 / total.cnt_all, 3) AS decimal(9,6))
AS float)
FROM
      total, cnt,
      hh.city c
      -- условие соединения по id городов и
      -- соответствию профессий (нижний регистр)
      JOIN hh.candidate cand ON cand.city_id = c.id
      AND lower(current_occupation) = lower(desirable_occupation)
GROUP BY c.title, total.cnt_all
 )
-- сортировка по количеству в обратном порядке
ORDER BY "Количество" desc
```

Data Output | Explain | Messages | Notifications

	Город text	Количество bigint	Доля в процентах (%) double precision
1	ВСЕГО КАНДИДАТОВ:	44744	100
2	Количество кандидатов с совпадающими должностями:	5371	12.004
3	Москва	1979	4.423
4	Санкт-Петербург	559	1.249
5	Новосибирск	126	0.282
6	Краснодар	122	0.273
7	Казань	108	0.241
8	Екатеринбург	91	0.203
9	Самара	86	0.192
10	Нижний Новгород	78	0.174

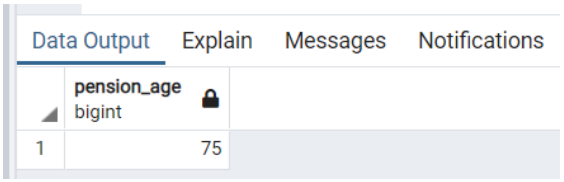
Теперь таблица стала более наглядной. Как и предполагалось, на первых строчках находятся Москва и Санкт-Петербург. Это объясняется тем, что, как установлено выше, Москва и Санкт-Петербург имеют самое большое количество кандидатов. Всего кандидатов с совпадающими должностями 12% – это не так много. Города, где количество кандидатов от 10 человек и ниже, имеют вовсе микроскопическую величину: тысячные доли процента. Сейчас развитие *IT*-индустрии идёт гигантскими шагами, появляются новые специальности в этой области и можно предположить, что в дальнейшем количество совпадающих текущей и желаемой профессий будет уменьшаться.

Задание 3.5

Определить количество кандидатов пенсионного возраста. Пенсионный возраст для мужчин наступает в 65 лет, для женщин – в 60 лет.

```
SELECT
    -- запрос на количество кандидатов
```

```
    count(cand.age) pension_age
FROM hh.candidate cand
WHERE
    -- фильтрация по возрасту и исключение
    -- "столетнего" выброса
    (age >= 60 AND gender = 'F')
    OR
    (age >= 65 AND gender = 'M')
    AND age < 100
```



Ответ:

Количество кандидатов пенсионного возраста (без “столетнего” выброса): 75 человек.

Вывод:

Полученное значение ничтожно мало. Неутешительное умозаключение – пенсионерам не до *IT*-работы. Большинство из них даже не знает, что существует такая отрасль. Сейчас пенсионерам “не до жиру”. Предполагаю, что если сравнить по данному вопросу пенсионеров России и, например, какой-либо европейской страны, результат для нас будет плачевным.

4. АНАЛИЗ КАНДИДАТОВ ДЛЯ ЗАКАЗЧИКА

Задание 4.1

Для добывающей компании нам необходимо подобрать кандидатов из Новосибирска, Омска, Томска и Тюмени, которые готовы работать вахтовым методом.

Формат выборки: *gender*, *age*, *desirable_occupation*, *city*, *employment_type*, *timetable_type*.

Отсортируйте результат по городу и номеру кандидата.

```
SELECT
      -- запрос по указанным в задании столбцам
      cand.gender,
      cand.age,
      cand.desirable_occupation,
      c.title city,
      cand.employment_type,
      tt.title timetable_type
FROM
      hh.candidate cand
      -- формирование join с необходимыми условиями соединения
      JOIN hh.city c ON cand.city_id = c.id
      JOIN hh.candidate_timetable_type ctt ON cand.id = ctt.candidate_id
      JOIN hh.timetable_type tt ON  ctt.timetable_id = tt.id
WHERE
      -- фильтрация по городам и вахтовому методу
      (c.title = 'Новосибирск'
      OR c.title = 'Омск'
      OR c.title = 'Томск'
      OR c.title = 'Тюмень')
      AND tt.title = 'вахтовый метод'
```

```
-- сортировка по city и id
ORDER BY city, cand.id
```

Data Output | Explain | Messages | Notifications

	gender character	age smallin	desirable_occupation text	city text	employment_type text	timetable_type... text
1	M	29	ИТ Инженер	Новосибирск	полная занятость	вахтовый ме...
2	M	25	Заместитель начал...	Новосибирск	проектная работа...	вахтовый ме...
3	M	30	Ведущий инженер, ...	Новосибирск	частичная занято...	вахтовый ме...
4	M	23	Программист	Новосибирск	полная занятость	вахтовый ме...
5	M	35	Инженер АСУТП, ин...	Омск	полная занятость	вахтовый ме...
6	M	25	Тестировщик ПО	Омск	стажировка, полн...	вахтовый ме...
7	M	26	Специалист технич...	Томск	частичная занято...	вахтовый ме...
8	M	30	Менеджер проектов	Томск	проектная работа...	вахтовый ме...
9	M	42	Инженер	Томск	проектная работа...	вахтовый ме...
10	M	31	Инженер связи	Тюмень	полная занятость	вахтовый ме...
11	M	31	Инженер АСУ ТП, А...	Тюмень	полная занятость	вахтовый ме...

Вывод:

Возраст *IT*-вахтовиков от 23 до 42 — самый расцвет сил и опыта. Их текущая профессия самая разнообразная и не является каким-то ключом, из-за которого человек пошёл на вахтовый метод работы. Пол кандидатов мужской, видимо, женщины опасаются вахтового метода. И не только. Если обратить внимание на общее количество данных кандидатов, то их из трёх крупных городов всего 11 человек. На мой взгляд — это малая величина.

Задание 4.2

Для заказчиков из Санкт-Петербурга необходимо собрать список из 10 желаемых профессий кандидатов из того же города от 16 до 21 года (в выборку включается 16 и 21, сортировка производится по возрасту) с указанием их возраста, а также добавить строку Total с общим количеством таких кандидатов. Составить запрос, который позволит получить выборку вида:

desirable_occupation	age
Системный администратор	16
Junior Разработчик C++/C#	18
3D-дизайнер	18
Unity3D developer Junior/middle	18
Специалист по IT	18
Java-разработчик	18
Программист	18
Руководитель web-разработки	18
HTML-верстальщик	18
Junior Data Scientist	18
Total	88

```
(SELECT
        -- запрос по желаемой профессии и возрасту
      cand.desirable_occupation,
      cand.age
```

```
FROM
      hh.city c
      -- join с поставленными условиями задачи
      JOIN hh.candidate cand ON c.id = cand.city_id
       AND c.title = 'Санкт-Петербург'
       AND (cand.age BETWEEN 16 AND 21)
-- сортировка по возрасту
ORDER BY cand.age
LIMIT 10)
-- объединение с нижней таблицей
UNION ALL
(SELECT
    -- формирование Total и запрос на общее количество
      'Total',
      COUNT(cand.age)
FROM
    hh.candidate cand,
    hh.city c
-- фильтрация согласно поставленным условиям задачи
WHERE (cand.age BETWEEN 16 AND 21)
      AND (c.title = 'Санкт-Петербург')
      AND (c.id = cand.city_id))
```

Data Output | Explain | Messages | Notifications

	desirable_occupation text	age bigint
1	Системный администратор	16
2	Junior Разработчик C++/C#	18
3	Программист	18
4	Junior Data Scientist	18
5	Руководитель web-разработки	18
6	Специалист по IT	18
7	Unity3D developer Junior/middle	18
8	HTML-верстальщик	18
9	3D-дизайнер	18
10	Java-разработчик	18
11	Total	161

Вывод:

Из 161-го кандидата в возрасте от 16 до 21 года в выборку, согласно условиям задания, попали первые десять желаемых профессий. Как видно, профессии совершенно разные, нет какой-то одной доминирующей. По возрасту можно считать, что кандидатам 18 лет. 16-летнего кандидата можно трактовать и как выброс, и как реальные данные, поэтому его оставляем.

ОБЩИЕ ВЫВОДЫ ПО ПРОЕКТУ

Дополнительно сделаю несколько запросов, а выводы по полученным результатам сразу включу в общий обзор по проекту.

Ранее в результатах не фигурировала зарплата кандидатов. Поэтому сделаю следующий запрос:

Вывести *id*-кандидатов, текущую должность, текущую зарплату

Первая сортировка по столбцу с должностями. Фиксация результата.

Вторая сортировка по столбцу с зарплатой по возрастанию.

```
SELECT
      c.id "id кандидатов",
      c.current_occupation "Текущая должность",
      c.salary "Текущая зарплата"
FROM hh.candidate c
ORDER BY 2 -- затем order by 3
```

Data Output | Explain | Messages | Notifications

	id кандидатов integer	Текущая должность text	Текущая зарплата numeric
1	15586	-	160000
2	17294	-	110000
3	5359	-	50000
4	36182	-	90000

С сортировкой order by 3:

Data Output | Explain | Messages | Notifications

	id кандидатов integer	Текущая должность text	Текущая зарплата numeric
1	35017	Менеджер проектов	1
2	43386	Исполнительный директор	1
3	36673	Инженер-программист	1
4	6992	Администратор сайта	1

Очередной запрос с участием зарплаты:

1. Размер текущей зарплаты кандидата должен лежать в диапазоне от 1.000.000 (вкл.) и выше.
2. В столбце “Текущая должность” вывести профессии кандидатов на данный момент.
3. В столбце “Текущая зарплата” вывести зарплату кандидатов на данный момент.
4. В начале таблицы, в первой строке, вывести количество кандидатов, согласно условию, указанном в 1.
5. Сортировать зарплату по убыванию.

```
(SELECT
       -- формирование столбцов, подсчёт количества кандидатов
        'Всего кандидатов с текущей зарплатой от 1000000 и выше:' "Текущая
должность",
       COUNT(*) "Текущая зарплата"
FROM hh.candidate can
-- фильтрация по заданному условию
WHERE can.salary >= 1000000
)
-- объединение с последующим запросом
UNION ALL
(SELECT
    -- запрос текущих должностей и зарплаты
    cand.current_occupation,
    cand.salary
FROM hh.candidate cand
```

```
-- фильтрация по заданному условию
WHERE salary >= 1000000
ORDER BY 2 desc)
```

Data Output | Explain | Messages | Notifications

	Текущая должность text	Текущая зарплата numeric
1	Всего кандидатов с текущей зарплатой от 1000000 и выше:	544
2	Директор	8000000
3	Директор по развитию	8000000
4	Директор по информационной безопасности (CISO)	8000000
5	senior asp.net developer	7000000

Теперь проведу этот же запрос, только с условием зарплаты кандидатов начиная от 10.000 и ниже.

Data Output | Explain | Messages | Notifications

	Текущая должность text	Текущая зарплата numeric
1	Всего кандидатов с текущей зарплатой от 10000 и ниже:	561
2	Ведущий разработчик	1
3	Менеджер проектов	1
4	Директор по продукту	1
5	начальник отдела Huawei Enterprise	1

Проведу выборку количества кандидатов от желаемой профессии:

```
SELECT
      count(c.desirable_occupation) "Количество",
      c.desirable_occupation "Желаемая должность"
FROM hh.candidate c
GROUP BY 2
ORDER BY 1 desc
```

Data Output | Explain | Messages | Notifications

	Количество bigint	Желаемая должность text
1	3099	Системный администратор
2	966	Инженер
3	809	Аналитик
4	745	Менеджер проектов
5	718	Руководитель проекта

И последний запрос, выводящий количество кандидатов по половому признаку.

```
SELECT
	count(c.gender) "Total",
	count(CASE WHEN c.gender = 'M' THEN 'M' END) "Man",
	count(CASE WHEN c.gender = 'F' THEN 'F' END) "Woman",
	 round((count(CASE WHEN c.gender = 'M' THEN 'M' END)) * 100.0 /
count(c.gender), 2) "% man",
	 round((count(CASE WHEN c.gender = 'F' THEN 'F' END)) * 100.0 /
count(c.gender), 2) "% woman"
FROM hh.candidate c
```

Data Output | Explain | Messages | Notifications

	Total bigint	Man bigint	Woman bigint	% man numeric	% woman numeric
1	44744	36211	8533	80.93	19.07

После проделанной работы можно сделать предварительный общий вывод по проекту. Дополнительно хочу заметить, что все запросы делались на неочищенных данных. Кроме нескольких раз, когда исключался “столетний” выброс.

КАНДИДАТЫ.

Общее число кандидатов — 44744. Из них женщин: 8533 (19.07%), мужчин: 36211 (80.93%). Кандидатов-мужчин больше, чем кандидатов-женщин, но я думаю, что это вопрос времени. И если сейчас процент женщин, ищущих *IT*-специальность невысок, то в дальнейшем будет расти.

Разделение количества кандидатов по городам показала явного лидера — Москву. С большим отрывом от неё идут Санкт-Петербург и Краснодар. Чем малочисленней город и дальше от центра, тем меньшее количество кандидатов.

Самая популярная желаемая должность — это “Системный администратор”. 3099 кандидатов. За ней идёт “Инженер”, “Аналитик”. Аналитики на третьем месте, но здесь это говорит не о спросе на них, а о том, что, возможно, их не хватает на рынке труда.

Много кандидатов у которых желаемая должность совпадает с текущей. 5371 человек или 12%, что является немалым результатом из которого можно допустить, что человек хочет, оставаясь на прежней должности, повысить зарплату, улучшить возможность карьеры и т.п.

Анализ на максимальный и минимальный возраст показал max - 100 лет, min - 14 лет. Я полагаю, что оба этих значения являются выбросами. Группировка по возрастам конкретно выделила лидирующую группу от 20 до 40 лет. На неё приходится 84.5% от общего количества кандидатов. На более зрелый возраст приходится меньше 14%.

Интересные значения дают запросы по зарплате. Так, при выводе зарплаты выше 1000000 получаем максимальную зарплату 8000000. Три позиции, все директора. Ведущий инженер, например, получает зарплату 7000000. Всего кандидатов с зарплатой выше 1000000 — 544 человека. Если изменить условие на зарплату меньше 10000, то получим зарплату… в 1 рубль! Причём, 12 позиций. Всего кандидатов с зарплатой меньше 10000 — 561 человек.

ОЧИСТКА ДАТАСЕТА.

Явные выбросы в возрастной области. 14, 76, 77, 100 лет: всего по 1 кандидату. Необходимо чистить.

Запрос по текущей должности выводит нам строки с прочерком. Если сортировать по убыванию, то первой строкой стоит поле со значением *null*. Названия должностей: “я, чем я занимаюсь, эникейщик”. Это явные “шутки”, уже не выбросы, а глупые вбросы недалёких людей. Такие записи надо выявлять и удалять.

Перейдём к запросу зарплаты. Зарплаты свыше 1000000 в количестве 544 кандидатов. Большинство из этих записей - аномалии. Также и зарплаты ниже 10000 в 1 рубль. Всё

это необходимо очищать. Если датасет не битый, то он подлежит тщательной проверке на предмет аномалий и выбросов с последующей очисткой.