Проект: Анализ вакансий на HH.ru с использованием SQL

# Знакомство с датасетом

В рамках данного проекта используется первоначальный датасет из предыдущего проекта ([Проект 1: Проект по базовому разведывательному анализу и чистке данных в датасете соискателей сайта HH.ru](https://github.com/thetitorenko/sf_data_science/tree/main/project_1)), которой был переделан в реляционную базу данных, состоящую из нескольких таблиц.

В рамках проекта используются следующие таблицы, расположенные в схеме **hh**:

* **hh.candidate**

Таблица хранит в себе общие данные по кандидатам:

* + ***id*** *- ID соискателя [внешний ключ на* ***hh.candidate\_timetable\_type****]*
  + **gender** - пол *[varchar]*
  + **age** - возраст *[int4]*
  + **desirable\_occupation** - желаемая должность *[varchar]*
  + ***city\_id*** *- город [внешний ключ на* ***hh.city****]*
  + **employment\_type** - вид занятости *[varchar]*
  + **current\_occupation** - текущая должность *[varchar]*
  + **updated\_at** - дата обновления записи *[date]*
  + **salary** - зарплата *[num]*
* **hh.city**

Таблица хранит код города и его название:

* ***id*** *- ID города [внешний ключ на hh.candidate]*
* **title** - названия города *[varchar]*
* **hh.candidate\_timetable\_type**

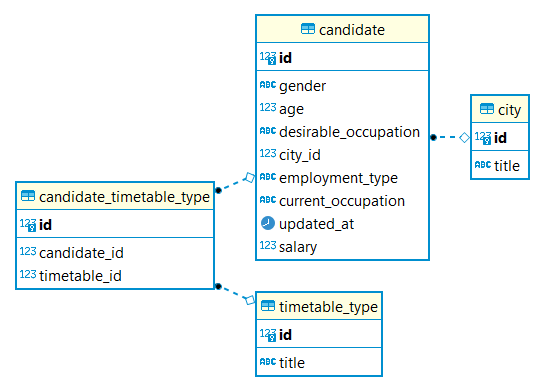
Таблица предназначена для организации связи многие-ко-многим, так как у нас есть много кандидатов и у них может быть несколько подходящих типов рабочего графика:

* ***id*** *- ID [первичный ключ]*
* ***candidate\_id*** *- ID соискателя [внешний ключ на* ***hh.candidate****]*
* ***timetable\_id*** *- ID рабочего графика [внешний ключ на* ***hh.timetable\_type****]*
* **hh.timetable\_type**

Таблица-справочник вариантов рабочего графика, подходящего кандидату:

* ***id*** *- ID рабочего графика [внешний ключ на* ***hh.candidate\_timetable\_type****]*
* **title** - вид рабочего графика *[varchar]*

ER-диаграмма, отображающая связь между таблицами схемы:



# Предварительный анализ данных

## Задание 2.1

Рассчитайте максимальный возраст (**max\_age**) кандидата в таблице.

Код:

| SELECT  max(age) max\_age FROM hh.candidate c |
| --- |

Результат**\***:



**\*** *здесь и далее - если результат целиком значит вывод SQL-запроса полный, если нет, то вывод обрезан и представлено только 10 первых значений*

Вывод:

Данный возраст имеет лишь один кандидат в таблице и, скорее всего, является выбросом и нуждается в удалении на этапе очистки данных. Пока его просто будем фильтровать в следующих запросах к БД.

## Задание 2.2

Рассчитайте минимальный возраст (**min\_age**) кандидата в таблице.

Код:

| SELECT  min(age) min\_age FROM hh.candidate c |
| --- |

Результат:



Вывод:

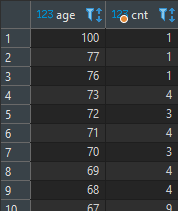
Минимальное значение возраста укладывается в рамки [законодательства РФ](http://www.ukn22.ru/so-skolki-let-mozhno-rabotat-v-rossii-uslovija/). Подростки, которым исполнилось 14 лет, могут сами подписывать договор, если родители или органы опеки дают свое согласие. Фильтровать запись не будем, т.к таблице присутствует лишь один кандидат с данным возрастом.

## Задание 2.3

Напишите запрос, который позволит рассчитать для каждого возраста (**age**) сколько (**cnt**) человек этого возраста у нас есть. Отсортируйте результат по возрасту в обратном порядке.

| SELECT  c.age,  count(\*) cnt *-- подсчет количества повторений уникального признака* FROM hh.candidate c GROUP BY 1 *-- получение уникальных значений признака возраста* ORDER BY 1 DESC *-- сортировка по возрасту в обратном порядке* |
| --- |

Результат:



Вывод:

Модальное значение возраста примерно равно [30-ти годам](https://github.com/thetitorenko/sf_data_science/blob/71a1cd77c4db546cdc8afc4e88163c7406466c2a/project_1/plotly_html/plotly_1_2.html). Возможно, это вызвано тем, что после окончания университета люди склонны идти работать по распределению, на кафедру и т.п. В более старшем же возрасте у людей уже, возможно, наработаны связи, поэтому пользоваться HeadHunter им необходимости нет. К 30 годам очень часто люди склонны сменить работу или вообще род деятельности, но достаточного опыта и востребованности пока еще не имеют, поэтому пользуются такими сервисами для поиска работы.

## Задание 2.4

Найдите количество кандидатов, которые старше среднего возраст занятых в экономике России (~39.7 лет под данным Росстата).

Код:

| SELECT  count(\*) cnt\_above\_avg *-- подсчет кол-ва соискателей после* фильтрации FROM hh.candidate c WHERE c.age BETWEEN 41 AND 99 *-- фильтрация по возрасту искл. < 40 и > 100* |
| --- |

Результат:



Вывод:

Если провести дополнительное исследование (см. ниже), то выясниться что соискателей в диапазоне выше среднего (41-99 лет) всего 14% от общего числа. Вероятно в этом возрасте люди уже менее склонны менять работу и вид деятельности.

| *-- создаем таблицу с кол-вом соискателей в каждом возрасте* WITH t AS( SELECT  c.age,  count(DISTINCT c.id) cnt FROM hh.candidate c GROUP BY c.age) SELECT  'Всего соискателей' applicant\_groups,  sum(t.cnt) cnt,  '100%' ratio *-- выводим общее кол-во соискателей* FROM t UNION ALL SELECT  'В возрасте 14-40 лет',  sum(t.cnt),  round((sum(t.cnt) \* 100 / (SELECT sum(t.cnt) FROM t)))::text||'%'  *-- считаем долю соискателей в возрасте 14-40 лет* FROM t WHERE t.age BETWEEN 14 AND 40 *-- фильтрация по возрасту* UNION ALL SELECT  'В возрасте 41-99 лет',  sum(t.cnt),  round((sum(t.cnt) \* 100 / (SELECT sum(t.cnt) FROM t)))::text||'%'  *-- считаем долю соискателей в возрасте 41-99 лет* FROM t WHERE t.age BETWEEN 41 AND 99 *-- фильтрация по возрасту* |
| --- |



# Глобальный анализ показателей

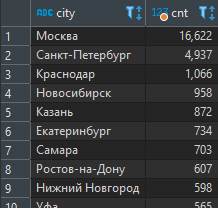
## Задание 3.1

Напишите запрос, который позволит узнать, сколько (**cnt**) у нас кандидатов из каждого города (**city**). Группировку таблицы необходимо провести по столбцу **title**, результат отсортируйте по количеству в обратном порядке.

Код:

| SELECT  c2.title city,  count(\*) cnt FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города* GROUP BY 1 *-- получение уникальных значений признака названия города* ORDER BY 2 DESC *-- результат отсортирован по количеству в обратном порядке* |
| --- |

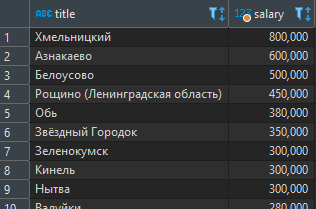
Результат:



Вывод:

Можно сделать вывод, что Москва самый крупный и активный рынок труда. Однако если взять зарплаты не выше 800 тыс. рублей (ЗП выше уже можно отнести к позициям топ-менеджмента и нецелевой группы пользователей сайта hh.ru), то картина измениться. Вакансий в Москве хоть и много, но по уровню зарплаты они не самые высокие.

| SELECT  c2.title,  round(avg(c.salary)) salary *-- считаем среднюю ЗП* FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города* WHERE c.salary <= 800000 *-- фильтрация ЗП не более 800 тыс. руб.* GROUP BY 1 *-- группировка по названию города* ORDER BY 2 DESC, 1 *-- результат отсортирован по уменьшению ЗП и городу* |
| --- |



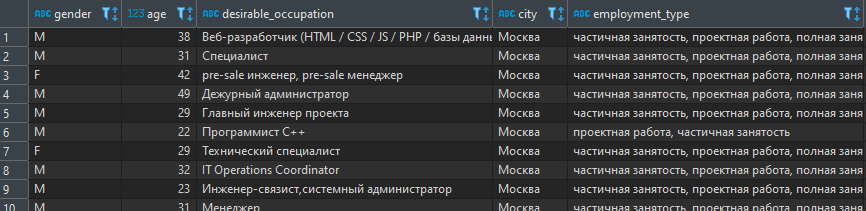
## Задание 3.2

Напишите запрос, который позволит понять, каких кандидатов из Москвы устроит «проектная работа». Формат выборки: **gender**, **age**, **desirable\_occupation**, **city**, **employment\_type**. Отсортируйте результат по **id** кандидата.

Код:

| SELECT  c.gender,  c.age,  c.desirable\_occupation,  c2.title city,  c.employment\_type FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города*  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  *-- соединение таблиц по внешнему ключу id соискателя*  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id  *-- соединение таблиц по внешнему ключу id рабочего графика* WHERE c2.title = 'Москва' AND c.employment\_type LIKE '%проектная работа%' *-- фильтрация соискателей проживающих в Москве и указавших нужный нам раб.график* ORDER BY c.id *-- результат отсортирован по id кандидата* |
| --- |

Результат:



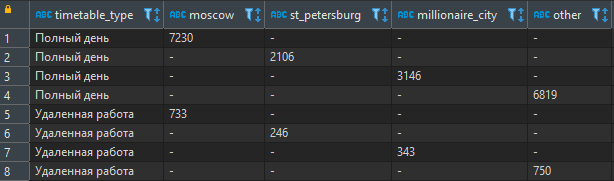
Вывод:

Кандидатов много – 2950 вакансий. Нужна дополнительная фильтрация (см. следующее задание).

В разрезе рабочего графика соискателя интересно также посмотреть насколько часто люди из регионов склонны работать удаленно. Для этого применим методологию из прошлого [проекта](https://github.com/thetitorenko/sf_data_science/tree/main/project_1), где мы создавали новые признаки на основании города проживания соискателя. Группировка будет аналогична: Москва, Санкт-Петербург, город-миллионник, другие (все кроме трех других).

Из результата ниже видно, что вакансий на удаленную работу в других городах даже больше чем в Москве, хотя на обычный полный график наблюдается обратная картина. Возможно люди не готовые к переезду и хотят работать удаленно в компаниях с других регионов и также рассчитывают на более высокую заработную плату.

| *-- таблицы по удаленной работе* SELECT  'Удаленная работа' timetable\_type,  '-' moscow,  '-' st\_petersburg,  count(\*)::text millionaire\_city, *-- подсчет количества*  '-' other FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города*  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  *-- соединение таблиц по внешнему ключу id соискателя*  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id  *-- соединение таблиц по внешнему ключу id рабочего графика* WHERE c2.title IN (  'Новосибирск', 'Екатеринбург', 'Нижний Новгород',  'Казань', 'Челябинск', 'Омск', 'Самара',  'Ростов-на-Дону', 'Уфа', 'Красноярск', 'Пермь',  'Воронеж', 'Волгоград')  *-- фильтрация по списку городов миллионников*  AND tt.title LIKE '%удаленная работа%'  *-- фильтрация по рабочему графику* UNION ALL SELECT  'Удаленная работа', count(\*)::text, '-', '-', '-' FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title = 'Москва' *-- фильтрация по г.Москва*  AND tt.title LIKE '%удаленная работа%' UNION ALL SELECT  'Удаленная работа', '-', count(\*)::text, '-', '-' FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title = 'Санкт-Петербург' *-- фильтрация по г.Спб*  AND tt.title LIKE '%удаленная работа%' UNION ALL  SELECT  'Удаленная работа', '-', '-', '-', count(\*)::text FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title NOT IN ('Москва', 'Санкт-Петербург',  'Новосибирск', 'Екатеринбург', 'Нижний Новгород',  'Казань', 'Челябинск', 'Омск', 'Самара',  'Ростов-на-Дону', 'Уфа', 'Красноярск', 'Пермь',  'Воронеж', 'Волгоград')  *-- фильтрация городов НЕ входящим в список городов миллионников*  AND tt.title LIKE '%удаленная работа%' UNION ALL *-- таблицы по полному дню* SELECT  'Полный день', '-', '-', count(\*)::text, '-' FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title IN (  'Новосибирск', 'Екатеринбург', 'Нижний Новгород',  'Казань', 'Челябинск', 'Омск', 'Самара',  'Ростов-на-Дону', 'Уфа', 'Красноярск', 'Пермь',  'Воронеж', 'Волгоград')  AND tt.title LIKE '%полный день%' UNION ALL SELECT  'Полный день', count(\*)::text "Москва", '-', '-', '-' FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title = 'Москва'  AND tt.title LIKE '%полный день%' UNION ALL SELECT  'Полный день', '-', count(\*)::text, '-', '-' FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title = 'Санкт-Петербург'  AND tt.title LIKE '%полный день%' UNION ALL  SELECT  'Полный день', '-', '-', '-', count(\*)::text FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c2.title NOT IN ('Москва', 'Санкт-Петербург',  'Новосибирск', 'Екатеринбург', 'Нижний Новгород',  'Казань', 'Челябинск', 'Омск', 'Самара',  'Ростов-на-Дону', 'Уфа', 'Красноярск', 'Пермь',  'Воронеж', 'Волгоград')  AND tt.title LIKE '%полный день%' ORDER BY 1, 5, 4, 3, 2 *-- сортировка по режиму работу и далее по городам* |
| --- |



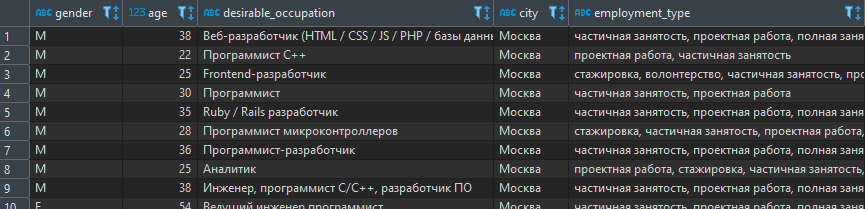
## Здание 3.3

Отфильтруйте только самые популярные IT-профессии — разработчик, аналитик, программист. Отсортируйте результат по **id** кандидата.

Код:

| SELECT  c.gender,  c.age,  c.desirable\_occupation,  c2.title city,  c.employment\_type FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города*  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  *-- соединение таблиц по внешнему ключу id соискателя*  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id  *-- соединение таблиц по внешнему ключу id рабочего графика* WHERE c2.title = 'Москва' AND c.employment\_type LIKE '%проектная работа%'  AND (lower(c.desirable\_occupation) LIKE '%разработчик%'  OR lower(c.desirable\_occupation) LIKE '%аналитик%'  OR lower(c.desirable\_occupation) LIKE '%программист%')  *-- фильтрация по вакансиям содержащие необходимые IT-профессии* ORDER BY c.id *-- результат отсортирован по id кандидата* |
| --- |

Результат:

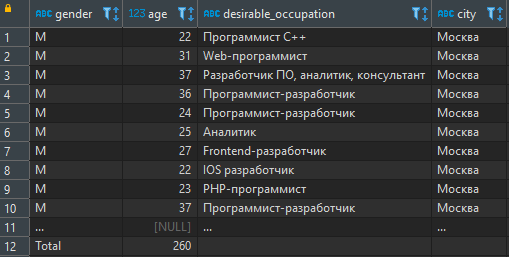


Вывод:

После фильтрации отсеялось 2172 вакансии. В текущей выборке присутствует 778 кандидата.

Общеизвестно, что люди IT-профессий склонны работать удаленно. Посмотрим, сколько будет таких человек.

| WITH t AS ( SELECT  c.gender,  c.age,  c.desirable\_occupation,  c2.title city FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города*  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  *-- соединение таблиц по внешнему ключу id соискателя*  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id  *-- соединение таблиц по внешнему ключу id рабочего графика* WHERE c2.title = 'Москва' AND tt.title LIKE '%удаленная работа%'  AND (lower(c.desirable\_occupation) LIKE '%разработчик%'  OR lower(c.desirable\_occupation) LIKE '%аналитик%'  OR lower(c.desirable\_occupation) LIKE '%программист%')  *-- фильтрация по вакансиям содержащие необходимые IT-профессии* ORDER BY c.id) *-- результат отсортирован по id кандидата* ) (SELECT \* FROM t LIMIT 10) UNION ALL SELECT  '...', NULL, '...', '...' UNION ALL SELECT  'Total', count(\*), ' ', ' ' FROM t |
| --- |



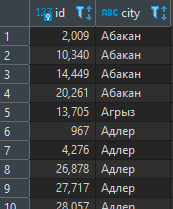
## Задание 3.4

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

Код:

| SELECT  c.id,  c2.title city FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города* WHERE c.desirable\_occupation = c.current\_occupation  *-- фильтрация по кандидатам у которых занимаемая должность аналогична желаемой* ORDER BY 2, 1 *-- результат отсортирован по городу и id кандидата* |
| --- |

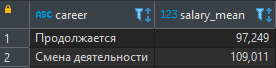
Результат:



Вывод:

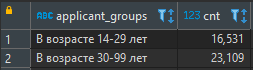
Ожидаем такие соискатели есть. Большинство из них предпочитают работать полный день, а также, что очень логично, имеют среднюю желаемую ЗП ниже, чем у соискателей желающих сменить род деятельности.

| *-- таблица с продолжающейся карьерой* SELECT  'Продолжается' сareer,  round(avg(c.salary)) salary\_mean *-- считаем среднюю ЗП* FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города*  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  *-- соединение таблиц по внешнему ключу id соискателя*  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id  *-- соединение таблиц по внешнему ключу id рабочего графика* WHERE c.desirable\_occupation = c.current\_occupation *-- фильтрация по совпадению занимаемой должности с желаемой* UNION ALL *-- таблица со сменной деятельности* SELECT  'Смена деятельности',  round(avg(c.salary)) FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id WHERE c.desirable\_occupation != c.current\_occupation *-- фильтрация по НЕсовпадению занимаемой должности с желаемой* |
| --- |



Развивая гипотезу из задания 2.3 о том, что к определенному возрасту (30 годам) люди склонны менять профессию, посмотрим у скольких соискателей занимаемая должность разница с желаемой. Добавим также фильтрацию по возрастным группам и агрегирующие функции для лучшей аналитики.

| SELECT  'В возрасте 14-29 лет' applicant\_groups,  count(\*) cnt FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id WHERE c.desirable\_occupation != c.current\_occupation  AND c.age BETWEEN 14 AND 29 UNION ALL  SELECT  'В возрасте 30-99 лет',  count(\*) FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id WHERE c.desirable\_occupation != c.current\_occupation  AND c.age BETWEEN 30 AND 99 |
| --- |



Определенная тенденция наблюдается, но для подтверждения гипотезы нужно больше данных.

## Задание 3.5

Определите количество кандидатов пенсионного возраста.

Код:

| *-- создаем соединенную таблицу для обращений* WITH t AS ( SELECT \* FROM hh.candidate c WHERE c.gender = 'M'  AND c.age BETWEEN 65 AND 99  *-- фильтрация по полу и возрасту* UNION ALL SELECT \* FROM hh.candidate c WHERE c.gender = 'F'  AND c.age BETWEEN 60 AND 99)  *-- фильтрация по полу и возрасту* SELECT  count(\*) retiree\_cnt  *-- подсчет количества соискателей (строк таблицы)* FROM t |
| --- |

Результат:



Вывод:

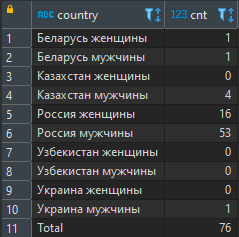
Количество таких людей очень мало и на общую статистику не влияет. Очевидно, что в пожилом возрасте люди не склонны искать работу, да и работодатели, к сожалению, мало заинтересованы в данной категории соискателей.

Однако если внимательнее изучить датасет, то можно увидеть, что география резюме довольно большая и охватывает много стран СНГ. В разных странах пенсионная система отличается возрастом.

Интересно посмотреть на количество пенсионеров в разрезе стран и их [пенсионной политики](https://e-cis.info/news/566/95094/) (отфильтруем людей по подготовленным спискам городов топ-5 стран СНГ по численности населения), а также на изменение их количества после.

Код подготовки списков городов (*Python*) и код скрипт SQL-запроса, ввиду количества строк, размещены в Github.

Результат запроса:



Общее количество пенсионеров изменилось на одного человека. С другой стороны можно заметить другую закономерность: мужчин пенсионного возраста явно больше женщин даже несмотря на пенсионную политику стран (возраст выхода мужчин на пенсию больше, чем у женщин). Можно предположить, что в осном женщины в данном возрасте больше заняты домашним хозяйством и присмотром и воспитанием внуков.

# Анализ кандидатов для заказчиков

## Задание 4.1

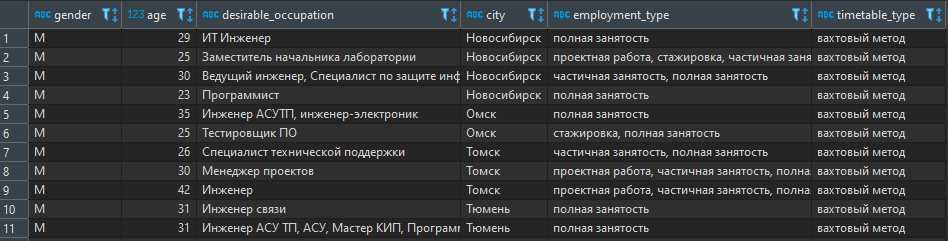
Для добывающей компании нам необходимо подобрать кандидатов из Новосибирска, Омска, Томска и Тюмени, которые готовы работать вахтовым методом.

Формат выборки: **gender**, **age**, **desirable\_occupation**, **city**, **employment\_type**, **timetable\_type**. Отсортируйте результат по городу и номеру кандидата.

Код:

| SELECT  c.gender,  c.age,  c.desirable\_occupation,  c2.title city,  c.employment\_type,  tt.title timetable\_type FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города*  LEFT JOIN hh.candidate\_timetable\_type ctt ON c.id = ctt.candidate\_id  *-- соединение таблиц по внешнему ключу id соискателя*  LEFT JOIN hh.timetable\_type tt ON ctt.timetable\_id = tt.id  *-- соединение таблиц по внешнему ключу id рабочего графика* WHERE c2.title IN ('Новосибирск', 'Омск', 'Томск', 'Тюмень')  AND tt.title LIKE '%вахтовый метод%'  *-- фильтрация соискателей по городу и рабочему графику* ORDER BY c2.title, c.id *-- результат отсортирован по городу и id соискателя* |
| --- |

Результат:



Вывод:

Все кандидаты мужского пола и среднего возраста 29,7 лет (**avg(c.age)**). Люда с данными параметрами обычно готовы к такому графику работы.

В выборке присутствуют соискатели с востребованными для нефтяной компании должностями. Однако общая выборка содержит всего 11 человек, что ограничивает выбор специалиста для специалиста по персоналу. Следует пересмотреть критерии фильтрации расширив список городов или/и рассмотреть возможность к найму людей готовых работать удаленно.

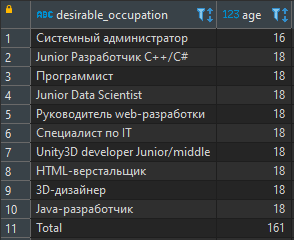
## Задание 4.2

Для заказчиков из Санкт-Петербурга нам 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.

Код:

| *-- таблица топ-10* (SELECT  c.desirable\_occupation,  c.age FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города* WHERE c.age BETWEEN 16 AND 21  AND c2.title = 'Санкт-Петербург'  *-- фильтрация по возрасту и городу соискателя* ORDER BY c.age *-- результат отсортирован по возрасту* LIMIT 10) *-- ограничение вывода 10 позициями* UNION ALL *-- соединение таблиц* *-- таблица с общим кол-вом* (SELECT  'Total',  count(\*) *-- подсчет общего количества соискателей после фильтрации* FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города* WHERE c.age BETWEEN 16 AND 21  AND c2.title = 'Санкт-Петербург')  *-- фильтрация по возрасту и городу соискателя* |
| --- |

Результат:



Вывод:

По общей картине результата запроса можно увидеть наличие большого числа IT-профессий. Сейчас данные профессии очень горячие и, соответственно, очень много только закончивших обучение молодых людей по данным направлениям. Если применить фильтрацию по популярным IT-профессиям из задания 3.3 (предварительно дополнив его еще несколькими профессиями), то можно понять их более точное количество.

| SELECT  count(\*) FROM hh.candidate c  JOIN hh.city c2 ON c.city\_id = c2.id  *-- соединение таблиц по внешнему ключу id города* WHERE c.age BETWEEN 16 AND 21  AND c2.title = 'Санкт-Петербург'  *-- фильтрация по возрасту и городу соискателя*  AND (lower(c.desirable\_occupation) LIKE '%разработчик%'  OR lower(c.desirable\_occupation) LIKE '%аналитик%'  OR lower(c.desirable\_occupation) LIKE '%программист%'  OR lower(c.desirable\_occupation) LIKE '%администратор%'  OR lower(c.desirable\_occupation) LIKE '%developer%'  OR lower(c.desirable\_occupation) LIKE '%it%'  OR lower(c.desirable\_occupation) LIKE '%тестировщик%')  *-- фильтрация по вакансиям содержащие необходимые IT-профессии* |
| --- |



# Общий вывод по проекту

На первых этапах знакомства и предварительного анализа данных мы понимаем границы возраста кандидатов, а также находим потенциальный выброс - возраст 100 лет. Нашли, что 6263 соискателя, старше среднего возраста занятых в экономике России людей, а также увидели количество людей в каждой возрастной группе (всего 63 группы).

Этап глобального анализа данных позволил глубже понять географию соискателей, найти самый крупный и активный рынок труда нашего датасета (г. Москва), сформировать выборку соискателей из Москвы готовых к определенному виду работы (проектная). В конце мы снова вернулись к временным рядам датасета и нашли количество пенсионеров в данных согласно указанному ими возраста.

В завершающем этапе проекта были подготовлены выборки под запросы компаний.

На каждом из этапов использовались основные функции SQL-запросов:

* Подсчет уникальных повторений, получение статистических данных
* Группировка по определенному признаку
* Фильтрация данных
* Сортировка данных
* Объединение нескольких таблиц датасета

SQL позволил взглянуть на старый датасет по-новому и позволил начать работать с данными и анализировать их в области Data science без дополнительного программирования.