

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](#)), 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:

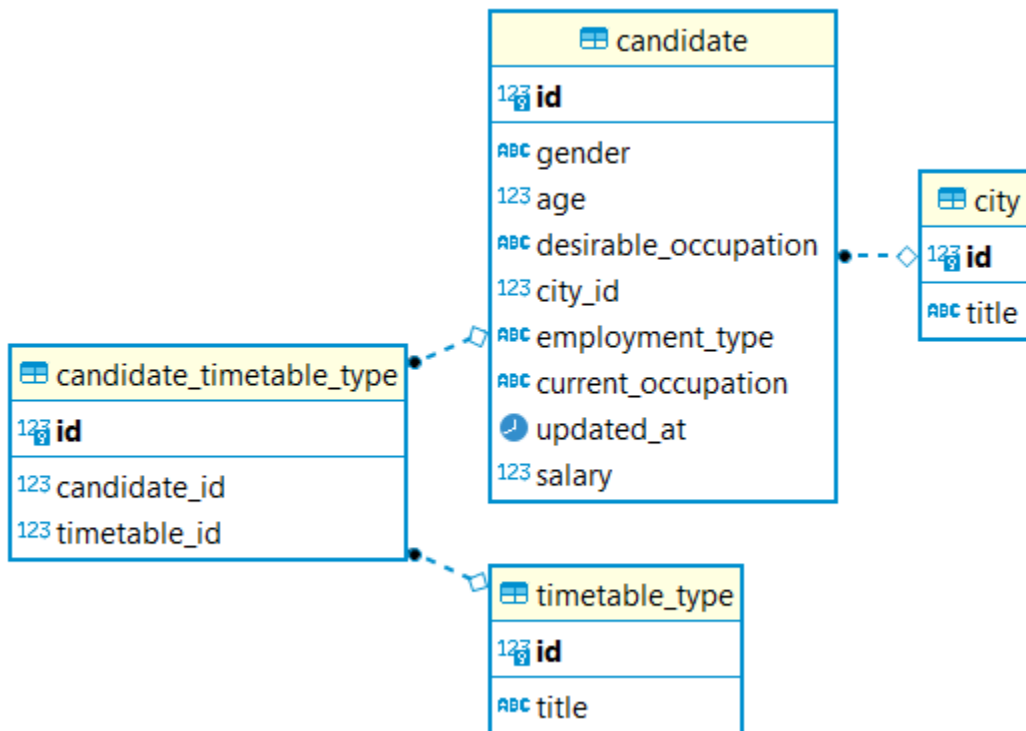
- *id* - ID [первичный ключ]
- *candidate_id* - ID соискателя [внешний ключ на *hh.candidate*]
- *timetable_id* - ID рабочего графика [внешний ключ на *hh.timetable_type*]

- **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*:

	123 max_age
1	100

* 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:

	123 min_age	
1		14

Вывод:

Минимальное значение возраста укладывается в рамки [законодательства РФ](#). 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:

	123 age T↑	123 cnt T↑
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:

Модальное значение возраста примерно равно [30-ти годам](#). 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:

	123 cnt_above_avg T↑
1	6,263

Вывод:

Если провести дополнительное исследование (см. ниже), то выяснится 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

```

	abc applicant_groups	123 cnt	abc ratio
1	Всего соискателей	44,744	100%
2	В возрасте 14-40 лет	38,480	86%
3	В возрасте 41-99 лет	6,263	14%

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:

	abc city	123 cnt
1	Москва	16,622
2	Санкт-Петербург	4,937
3	Краснодар	1,066
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:

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
```

	abc title	123 salary
1	Хмельницкий	800,000
2	Азнакаево	600,000
3	Белоусово	500,000
4	Рощино (Ленинградская область)	450,000
5	Обь	380,000
6	Звёздный Городок	350,000
7	Зеленокумск	300,000
8	Кинель	300,000
9	Нытва	300,000
10	Ватейки	280,000

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:

	abc gender	123 age	abc desirable_occupation	abc city	abc employment_type
1	M	38	Веб-разработчик (HTML / CSS / JS / PHP / базы данн	Москва	частичная занятость, проектная работа, полная заня
2	M	31	Специалист	Москва	частичная занятость, проектная работа, полная заня
3	F	42	pre-sale инженер, pre-sale менеджер	Москва	частичная занятость, проектная работа, полная заня
4	M	49	Дежурный администратор	Москва	частичная занятость, проектная работа, полная заня
5	M	29	Главный инженер проекта	Москва	частичная занятость, проектная работа, полная заня
6	M	22	Программист C++	Москва	проектная работа, частичная занятость
7	F	29	Технический специалист	Москва	частичная занятость, проектная работа, полная заня
8	M	32	IT Operations Coordinator	Москва	частичная занятость, проектная работа, полная заня
9	M	23	Инженер-связист, системный администратор	Москва	частичная занятость, проектная работа, полная заня
10	M	31	Менеджер	Москва	частичная занятость, проектная работа, полная заня

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](#), 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

```

	ABC timetable_type	ABC moscow	ABC st_petersburg	ABC millionaire_city	ABC other
1	Полный день	7230	-	-	-
2	Полный день	-	2106	-	-
3	Полный день	-	-	3146	-
4	Полный день	-	-	-	6819
5	Удаленная работа	733	-	-	-
6	Удаленная работа	-	246	-	-
7	Удаленная работа	-	-	343	-
8	Удаленная работа	-	-	-	750

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:

	abc gender	123 age	abc desirable_occupation	abc city	abc employment_type
1	M	38	Веб-разработчик (HTML / CSS / JS / PHP / базы данных)	Москва	частичная занятость, проектная работа, полная занятость
2	M	22	Программист C++	Москва	проектная работа, частичная занятость
3	M	25	Frontend-разработчик	Москва	стажировка, волонтерство, частичная занятость, проектная работа
4	M	30	Программист	Москва	частичная занятость, проектная работа
5	M	35	Ruby / Rails разработчик	Москва	частичная занятость, проектная работа, полная занятость
6	M	28	Программист микроконтроллеров	Москва	стажировка, частичная занятость, проектная работа, полная занятость
7	M	36	Программист-разработчик	Москва	частичная занятость, проектная работа, полная занятость
8	M	25	Аналитик	Москва	проектная работа, стажировка, частичная занятость, проектная работа
9	M	38	Инженер, программист C/C++, разработчик ПО	Москва	частичная занятость, проектная работа, полная занятость
10	F	54	Ведущий инженер-программист	Москва	частичная занятость, проектная работа, полная занятость

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
```

	ABC gender T↑	123 age T↑	ABC desirable_occupation T↑	ABC city T↑
1	M	22	Программист C++	Москва
2	M	31	Web-программист	Москва
3	M	37	Разработчик ПО, аналитик, консультант	Москва
4	M	36	Программист-разработчик	Москва
5	M	24	Программист-разработчик	Москва
6	M	25	Аналитик	Москва
7	M	27	Frontend-разработчик	Москва
8	M	22	IOS разработчик	Москва
9	M	23	PHP-программист	Москва
10	M	37	Программист-разработчик	Москва
11	...	[NULL]
12	Total	260		

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:

	id	city
1	2,009	Абакан
2	10,340	Абакан
3	14,449	Абакан
4	20,261	Абакан
5	13,705	Агрыз
6	967	Адлер
7	4,276	Адлер
8	26,878	Адлер
9	27,717	Адлер
10	28,057	Адлер

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' career,
    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 деятельностью
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
```

	abc career	123 salary_mean
1	Продолжается	97,249
2	Смена деятельности	109,011

Развивая гипотезу из задания 2.3 о том, что к определенному возрасту (30 годам) люди склонны менять профессию, посмотрим у скольких соискателей занимаемая должность разнится 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
```

	abc applicant_groups	123 cnt
1	В возрасте 14-29 лет	16,531
2	В возрасте 30-99 лет	23,109

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:

	123 retiree_cnt	
1		75

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](#) (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:

	abc country	123 cnt
1	Беларусь женщины	1
2	Беларусь мужчины	1
3	Казахстан женщины	0
4	Казахстан мужчины	4
5	Россия женщины	16
6	Россия мужчины	53
7	Узбекистан женщины	0
8	Узбекистан мужчины	0
9	Украина женщины	0
10	Украина мужчины	1
11	Total	76

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 основном 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:

	abc gender	123 age	abc desirable_occupation	abc city	abc employment_type	abc timetable_type
1	M	29	ИТ Инженер	Новосибирск	полная занятость	вахтовый метод
2	M	25	Заместитель начальника лаборатории	Новосибирск	проектная работа, стажировка, частичная заня	вахтовый метод
3	M	30	Ведущий инженер, Специалист по защите инф	Новосибирск	частичная занятость, полная занятость	вахтовый метод
4	M	23	Программист	Новосибирск	полная занятость	вахтовый метод
5	M	35	Инженер АСУТП, инженер-электроник	Омск	полная занятость	вахтовый метод
6	M	25	Тестировщик ПО	Омск	стажировка, полная занятость	вахтовый метод
7	M	26	Специалист технической поддержки	Томск	частичная занятость, полная занятость	вахтовый метод
8	M	30	Менеджер проектов	Томск	проектная работа, частичная занятость, полна	вахтовый метод
9	M	42	Инженер	Томск	проектная работа, частичная занятость, полна	вахтовый метод
10	M	31	Инженер связи	Тюмень	полная занятость	вахтовый метод
11	M	31	Инженер АСУ ТП, АСУ, Мастер КИП, Програми	Тюмень	полная занятость	вахтовый метод

Вывод:

Все кандидаты мужского пола и среднего возраста 29,7 лет ($\text{avg}(\text{c.age})$). Люди 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:

	abc desirable_occupation	123 age
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:

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

```

	123 count 
1	94

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 возрасту.

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.