**Выбор датасета.**

В качестве датасета для анализа и построения графа связей я выбрала социальную сеть ВКонтакте и своих друзей. Я решила воспользоваться именно этой базой данных по нескольким причинам. Во-первых, частое использование сети делает анализ более показательным и интересным для обработки. Во-вторых, данные предоставлены явно и информативны, а поэтому удобны для анализа и построения графов. В-третьих, анализ социальной сети ВКонтакте открывает возможности к проверке новых данных и построению новых графов. Например, можно увидеть следующую информацию:

* Количество моих друзей ВКонтакте, их общие друзья
* Информация о самих друзьях (нодах), либо в качестве списка признаков, либо также визуализированная на самом графе
* Кластера объединенных (близко расположенных друг к другу) друзей и их значения
* Возможные значения: объединение по организации, по месту проживания, итд
* Разные другие закономерности связей между людьми
* Подтверждение теории о 5 рукопожатиях, а именно:
  + Найти популярных людей, с кем я знакома через 1, 2, 3, 4 и 5 рукопожатий
  + Определить популярность человека через количество подписчиков, лайков, количество подарков ВКонтакте

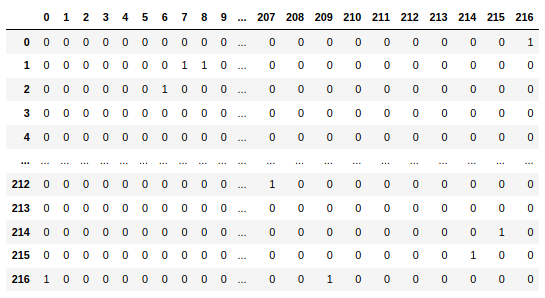
Как видно, есть множество возможностей для анализа и новых получения новых данных. Поэтому я и решила остановиться на выборе графа друзей ВКонтакте. Некоторые из предложенных выше пунктов предполагаются как дальнейшие пути развития.

**Сбор данных.**

Для сбора данных я написала свой код, а не брала готовый датасет. Чтобы достать данные о друзьях ВКонтакте и связях, я воспользовалась предоставленным ВК API. Использовались такие методы, как:

* friends.get()
* friends.getMutual()
* users.get()

Первым делом, я получила список своих друзей и общих друзей между ними. После чего составила матрицу смежности, отсортировав друзей по их id. Получилась следующая таблица (часть ее, pandas):



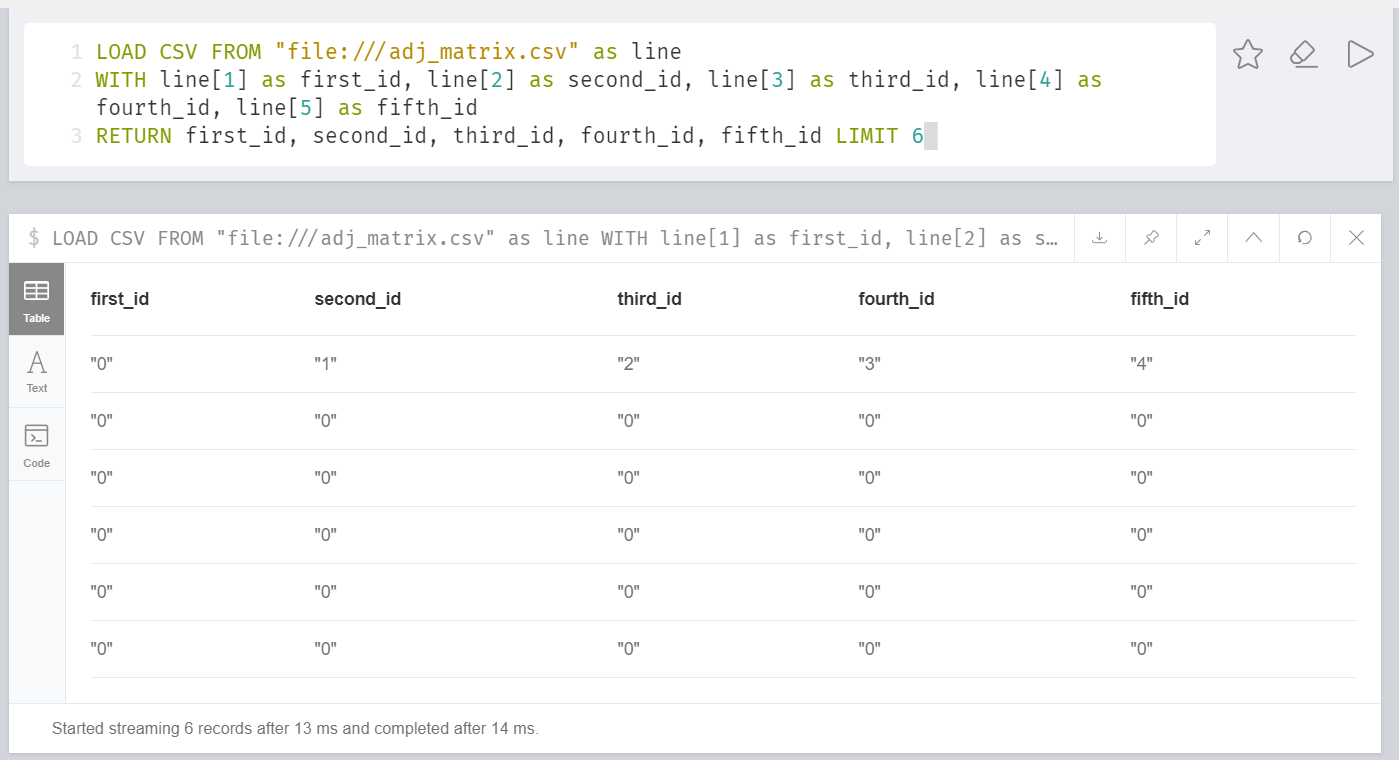
Далее с помощью API я достала информацию о своих друзьях и составила отдельную таблицу. А именно, были созданы следующие поля (если таковые имелись):

|  |  |
| --- | --- |
| name | Имя |
| last\_name | Фамилия |
| sex | Пол |
| bdate | Дата рождения |
| city | Город (первый указанный) |
| country | Страна (первая указанная) |
| has\_mobile | Наличие мобильной версии (бинарное) |
| status | Статус |
| followers\_count | Количество подписчиков (число) |
| nickname | Отчество |
| common\_count | Количество общих друзей (число) |
| has\_photo | Наличие фотографии в профиле (бинарное) |
| screen\_name | Ник (при его отсутствие - id) |

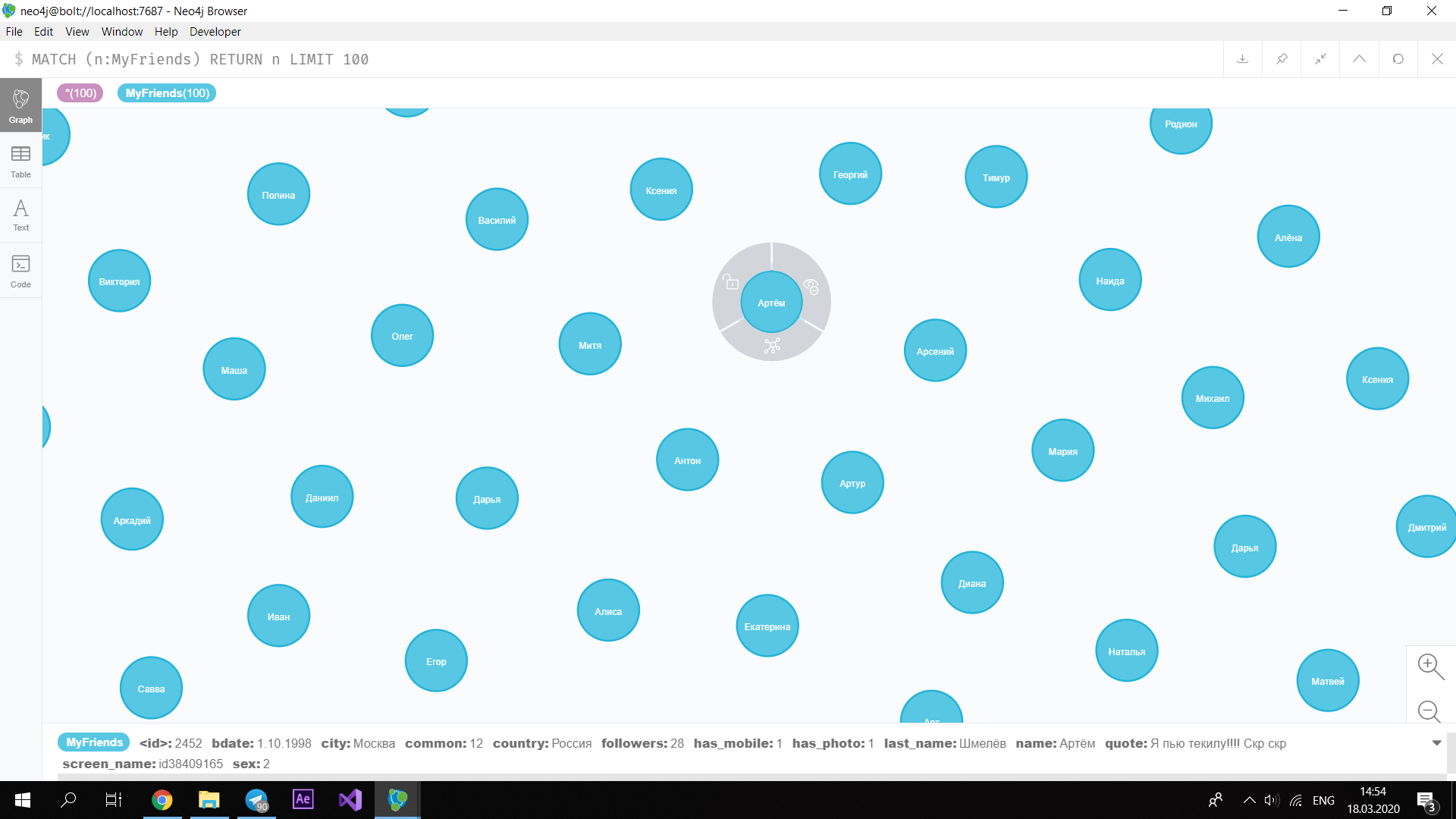
Таблица с результатами (pandas) выглядит так:



Полученные данные я загрузила в neo4j и визуализировала. Первое, я визуализировала матрицу смежности. Но из-за большого ее масштаба решила показать только маленькую ее часть (матрица 5\*5, хотя друзей гораздо больше). Полученная визуализация и ее запрос в neo4j:



Далее решила визуализировать таблицу с данными о друзьях. А именно - сразу построить граф. Однако, пока что без связей между нодами. Но чтобы каждая нода представляла собой полноценную информацию о человеке и о его признаках. Полученный граф (а точнее, его часть) представлен далее:



Далее я буду пробовать другие алгоритмы визуализации на моих данных и построю новые графы связей, либо дополню существующие.