
Investigating COVID Cases and Contact Tracing at UCSD

By: Harin Radha-Krishnan(A59019874), Swetha Arunraj(A59019948), Adhvaith Vijay

Introduction





Context

- California has 10,651,573 confirmed cases of COVID-19
 - San Diego alone accounting for 1.02M of these cases
- COVID-19 spread has been rampant across U.S. college campuses even in spite of weekly testing
 - More than 260,000 coronavirus cases have been linked to American universities
- COVID-19 data dashboards illustrate case rates among students, staff, and faculty
 - how can we take it one step further?

Taking it one step further

- In our post-COVID world where we have seen the brunt of the virus the question exists: What should we have done better?
 - Contact Tracing
- Using graph databases we can figure out which of a person's connections might be "high risk"
- We can analyze indirect contacts (i.e. between our friends' friends' friends)



Goals

Spot the COVID hotspots (areas where a lot of people are +ve and/or susceptible to COVID)

Contact tracing (family and colleagues of people tested +ve who are susceptible to COVID) using JSON data

Find details of people who are COVID positive and their families/ colleagues

Finding susceptible people and area based on their visiting time overlap with COVID +ve people

Use Cases described in the project:

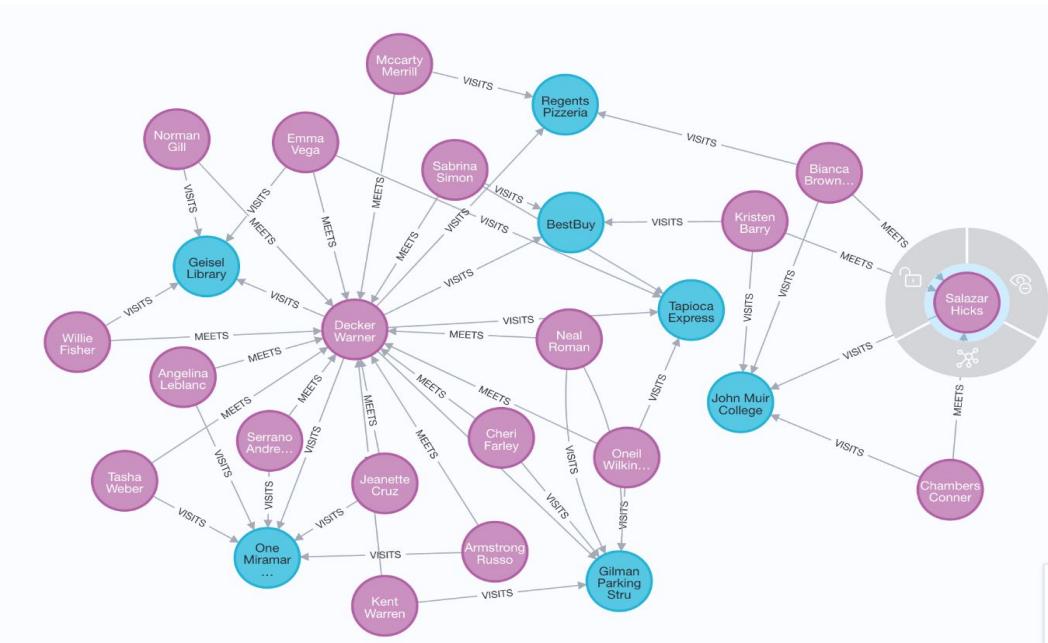
1. Retrieve information of people tested positive for COVID.
2. Analyse the susceptibility of COVID in each department, profession and resident status of UCSD.
3. Analyse the spread of COVID cases in the vicinity of UCSD using the longitudes and latitudes.
4. Retrieve information about the family members who are staying along with the COVID infected people.
5. Retrieve susceptible colleagues of COVID infected person who belonged to a particular Company.
6. People who went to the same place as that of the infected person at the same time duration to be found.
7. Non-Positive people who have a higher overlap time with the infected person to be found.
8. Places visited by people after being tested positive for covid to be found.
9. COVID hotspots to be detected.

Data Sources

Neo4j

- Graph database that takes input from loaded CSVs a person's COVID status and the place they visit
- We generate a network graph of interactions and locations
- We establish connections between people based on their visits

Role: The central database where the data of covid infected people and the places they visited are stored in a graph structure.



Data & structure in neo4j database

```
match (p:Results)-[v1:VISITS]→(pl:Places)
return p,v1,pl
limit 10
```

p	v1	pl
<pre>{ "identity": 3351, "labels": ["Results"], "properties": { "personName": "Mia Wong", "covidStatus": "Inconclusive", "confirmedTime": "2022-10-31T19:30:32Z", "personId": "34" }, "elementId": "3351" }</pre>	<pre>{ "identity": 16909, "start": 3351, "end": 3216, "type": "VISITS", "properties": { "exitTime": "2022-10-29T23:27:41Z", "entryTime": "2022-10-29T06:57:12Z", "visitId": "546" }, "elementId": "16909", "startNodeElementId": "3351", "endNodeElementId": "3216" }</pre>	<pre>{ "identity": 3216, "labels": ["Places"], "properties": { "placeId": "1", "placeName": "UC San Diego School of Medicine" }, "elementId": "3216" }</pre>
<pre>{ "identity": 3816, "labels": ["Results"], "properties": { "personName": "Briana Estrada", "covidStatus": "Not tested", "confirmedTime": "1900-01-01T00:00:01Z", "personId": "499" }, "elementId": "3816" }</pre>	<pre>{ "identity": 16620, "start": 3816, "end": 3216, "type": "VISITS", "properties": { "exitTime": "2022-10-29T19:57:34Z", "entryTime": "2022-10-29T00:53:55Z", "visitId": "257" }, "elementId": "16620", "startNodeElementId": "3816", "endNodeElementId": "3216" }</pre>	<pre>{ "identity": 3216, "labels": ["Places"], "properties": { "placeId": "1", "placeName": "UC San Diego School of Medicine" }, "elementId": "3216" }</pre>



PostgreSQL

- CSV data file that consists of personal information of each person associated with UCSD.
- The columns of the file are as given in the table
- Total number of records(rows): 1504

Role: The relational database is used to retrieve information of individuals infected by COVID and find the possible spread in the campus area and departments.

Column name	Data Type
ID	Integer
Name	varchar(32)
Mail_ID	varchar(32)
Phone_Number	varchar(18)
Acad_Status (Profession, Dept)	text
Resident_Status	text
AddressLat	numeric
AddressLong	numeric

K < 1-500 v of 501+ > | Tx: Auto | DDL | ↗

	<input type="checkbox"/> id	<input type="checkbox"/> name	<input type="checkbox"/> mail_id	<input type="checkbox"/> phone_number	<input type="checkbox"/> acad_status	<input type="checkbox"/> resident_status	<input type="checkbox"/> addresslat	<input type="checkbox"/> addresslong
1	1	Bentley Allison	bentleyallison@yomail.com	6537348001	PhD, Engineering	Off campus	32.685342	-117.200253
2	2	Mccray Schmidt	mccrayschmidt@yomail.com	9574849807	Undergraduate, Scripps Institution of Oceanography	On campus	32.916801	-117.1576
3	3	Hilda Henry	hildahenry@yomail.com	6054406181	PhD, Scripps Institution of Oceanography	Off campus	32.916609	-117.157861
4	4	Hoover Mathews	hoovermathews@yomail.com	9867295536	Undergraduate, Data Science	On campus	32.908906	-117.097802
5	5	Keri Medina	kerimedina@yomail.com	6787426993	Undergraduate, Arts and Humanities	On campus	32.906632	-117.107203
6	6	Hayes Marshall	hayesmarshall@yomail.com	6914782864	Masters, Scripps Institution of Oceanography	Off campus	32.882655	-117.233359
7	7	Aguirre Lawson	aguirrelawson@yomail.com	6402732953	Professor, Social Sciences	On campus	32.804875	-117.226054
8	8	Edwards Daniels	edwardsdaniels@yomail.com	7453140221	PhD, Social Sciences	Off campus	32.937189	-117.23668
9	9	Decker Warner	deckerwarner@yomail.com	9537203229	Undergraduate, Scripps Institution of Oceanography	On campus	32.699984	-117.132488
10	10	Lizzie Haney	lizziehaney@yomail.com	8838663520	Masters, Data Science	On campus	32.881032	-117.24464
11	11	Barnes Conley	barnesconley@yomail.com	8991229522	Undergraduate, Data Science	Off campus	32.697701	-117.245926
12	12	Rosanna Bryan	rosannabryan@yomail.com	7570667345	Masters, Data Science	On campus	32.775187	-117.179797
13	13	Salazar Hicks	salazarhicks@yomail.com	6226465268	PhD, Arts and Humanities	On campus	32.879848	-117.12107
14	14	Wong Malone	wongmalone@yomail.com	6842210211	Professor, Scripps Institution of Oceanography	On campus	32.744215	-117.133276
15	15	Brittney Blankenship	brittneyblankenship@yomail.com	6761506928	Professor, Arts and Humanities	On campus	32.85924	-117.135341
16	16	Morgan Compton	morgancompton@yomail.com	7902164474	Professor, Data Science	Off campus	32.768807	-117.125255
17	17	May Morse	maymorse@yomail.com	6543336179	Undergraduate, Biological Sciences	Off campus	32.715066	-117.142394
18	18	Leon Merritt	leonmerritt@yomail.com	7986284719	Professor, Data Science	On campus	32.729768	-117.184385
19	19	Kaye Sears	kayeseears@yomail.com	9438792982	PhD, Biological Sciences	Off campus	32.703692	-117.287191
20	20	Martina Riddle	martinariddle@yomail.com	8799105640	Staff, Social Sciences	Off campus	32.716022	-117.121554
21	21	Willie Meyer	williemeyer@yomail.com	8281190856	Professor, Engineering	On campus	32.886537	-117.145661
22	22	Yvette Lawrence	yvettelawrence@yomail.com	6794092919	PhD, Social Sciences	Off campus	32.887666	-117.194216
23	23	Tillman Dominguez	tillmandominguez@yomail.com	7637521911	Undergraduate, Social Sciences	Off campus	32.831671	-117.19574
24	24	Suarez Underwood	suarezunderwood@yomail.com	9258666707	Masters, Data Science	On campus	32.830656	-117.244889
25	25	Bartlett Fleming	bartlettfleming@yomail.com	8033948665	PhD, Data Science	On campus	32.896891	-117.167449
26	26	Velez McClure	velezmcclosure@yomail.com	6418438207	Masters, Biological Sciences	Off campus	32.777612	-117.187101
27	27	Erica Beach	ericabeach@yomail.com	7734896331	Masters, Scripps Institution of Oceanography	On campus	32.788213	-117.104954
28	28	Robbins Koch	robbinskoch@yomail.com	7396113843	Undergraduate, Arts and Humanities	On campus	32.955	-117.112762

Sample of the relational database



MongoDB

- JSON data consisting of the details of family members and colleagues for each person associated with UCSD.
- For each Person object, the properties are as given in the table.
- Total number of Person objects: 1504

Role: Used to retrieve information about the family members who live along with the infected person and the potential infection to their colleagues.

Property	Data Type
UniqueID	string
index	number
picture	string
name	string
address	string
latitude	number
longitude	number
Company	string
FamilyMembers	Array of objects (Each FamilyMembers object has id(number), name(string), age(number), Relationship(string), isStayingAlong(boolean))
Colleagues	Array of objects (Each Colleagues object has name(string), CompanyID(number), email(string), age(number))

Sample JSON Person objects with variable number of Family Members and Colleagues

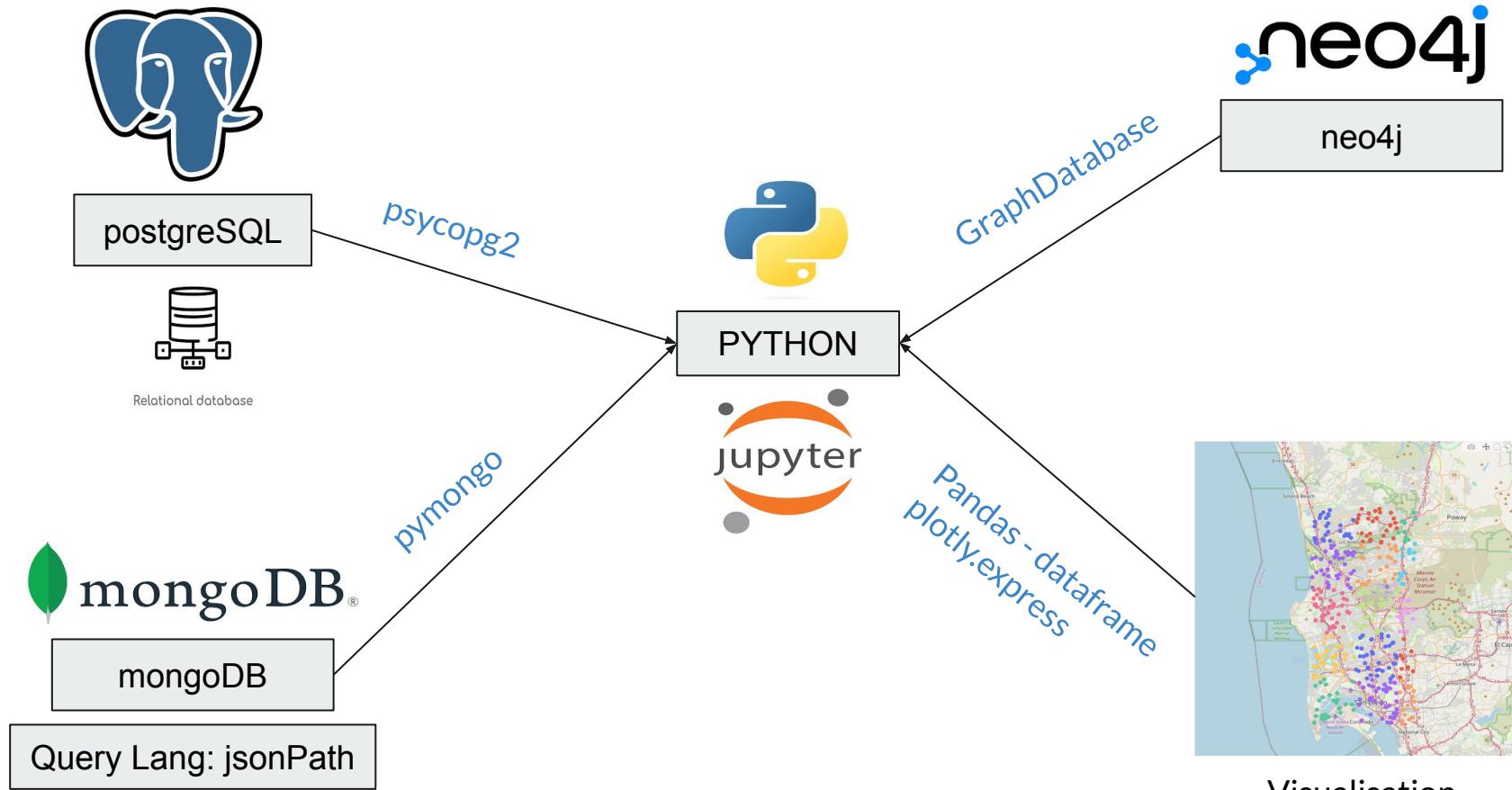
```
{
  "UniqueID": "63819d2bf8c2da035df1e74",
  "index": 1,
  "picture": "http://picSite.com/64x64",
  "name": "Bentley Allison",
  "address": "225 Brighton Court, Sims, Northern Mariana Islands, 3700",
  "latitude": 32.685342,
  "longitude": -117.200253,
  "Company": "BLEENDOT",
  "FamilyMembers": [
    {
      "id": 1,
      "name": "Charlotte Robles",
      "age": 34,
      "Relationship": "Sibling",
      "isStayingAlong": false
    },
    {
      "id": 2,
      "name": "Delgado Fernandez",
      "age": 28,
      "Relationship": "Parent",
      "isStayingAlong": false
    }
  ],
  "Colleagues": [
    {
      "name": "Nadia Paul",
      "CompanyID": 846766,
      "email": "nadiapaul@bleendot.com",
      "age": 55
    },
    {
      "name": "Connie Mathis",
      "CompanyID": 9166890,
      "email": "conniemathis@bleendot.com",
      "age": 47
    },
    {
      "name": "Henry Hale",
      "CompanyID": 1501311,
      "email": "henryhale@bleendot.com",
      "age": 56
    },
    {
      "name": "Raquel Salas",
      "CompanyID": 3532177,
      "email": "raquelsalas@bleendot.com",
      "age": 49
    },
    {
      "name": "Tonya Jordan",
      "CompanyID": 4984014,
      "email": "tonyajordan@bleendot.com",
      "age": 36
    },
    {
      "name": "Lacey Gibson",
      "CompanyID": 3359641,
      "email": "laceygibson@bleendot.com",
      "age": 21
    }
  ]
},
{
  "UniqueID": "63819d2bd6ada6936d6804bd",
  "index": 2,
  "picture": "http://picSite.com/64x64",
  "name": "McCray Schmidt",
  "address": "970 Strickland Avenue, Hatteras, Kentucky, 2033",
  "latitude": 32.916801,
  "longitude": -117.1576,
  "Company": "BUZZMAKER",
  "FamilyMembers": [
    {
      "id": 1,
      "name": "Cherie Holmes",
      "age": 17,
      "Relationship": "Child",
      "isStayingAlong": false
    },
    {
      "id": 2,
      "name": "Nixon Walls",
      "age": 46,
      "Relationship": "Sibling",
      "isStayingAlong": true
    },
    {
      "id": 3,
      "name": "Townsend Kent",
      "age": 60,
      "Relationship": "Grandparent",
      "isStayingAlong": false
    }
  ],
  "Colleagues": [
    {
      "name": "Ray Sharpe",
      "CompanyID": 8217843,
      "email": "raysharpe@buzzmaker.com",
      "age": 34
    }
  ]
},
{
  "UniqueID": "63819d2b9e5932c0b18ab0f0",
  "index": 3,
  "picture": "http://picSite.com/64x64",
  "name": "Hilda Henry",
  "address": "112 Irving Street, Albrightsville, Kansas, 3308",
  "latitude": 32.916609,
  "longitude": -117.157861,
  "Company": "JETSILK",
  "FamilyMembers": [
    {
      "id": 1,
      "name": "Harrell Cain",
      "age": 16,
      "Relationship": "Child",
      "isStayingAlong": true
    },
    {
      "id": 2,
      "name": "Roman Myers",
      "age": 71,
      "Relationship": "Child",
      "isStayingAlong": false
    }
  ]
}
```

Methodology:



Tools and Libraries used:

- Python (Jupyter Notebook)
- Neo4j - GraphDatabase
- Postgresql - psycopg2
- MongoDB - pymongo (MongoClient)
- jsonpath-ng - jsonpath, parse
- pandas - dataframe
- re
- plotly.express



"ID" and "name" of each Person are unique and "name" is used to connect all the data sources

Demo

- Data of people tested positive for COVID retrieved from neo4j database, and passed on to Postgres as a table from CSV file

```
query1 = """
    match(p1:Results)
    WHERE p1.covidStatus="Positive"
    return p1.personName as PersonName;
"""
```

```
results = session.run(query1)
data = results.data()
df = pd.DataFrame(data)
df.to_csv('neo4j_results.csv')
print(df)
```

	PersonName
0	Decker Warner
1	Salazar Hicks
2	Wong Malone
3	Kaye Sears
4	Velez Mcclure
..	...
404	Schultz Bright
405	Joan Brennan
406	Chase Hutchinson
407	Paul Duncan
408	Le Benton

Relational Database

```
SELECT id, name, mail_id, phone_number  
FROM person p  
RIGHT JOIN neo4j_results n  
ON p.name=n.personname
```

Basic query to retrieve information of people who tested +ve for COVID

Output postgres.public.person

	id	name	mail_id	phone_number
1	9	Decker Warner	deckerwarner@yomail.com	9537203229
2	13	Salazar Hicks	salazarhicks@yomail.com	6226465268
3	14	Wong Malone	wongmalone@yomail.com	6842210211
4	19	Kaye Sears	kayeseears@yomail.com	9430792982
5	26	Velez Mcclure	velezmcclure@yomail.com	6418438207
6	28	Robbins Koch	robbinskoch@yomail.com	7396113843
7	37	Mildred Mullins	mildredmullins@yomail.com	9704112986
8	40	Jenifer White	jeniferwhite@yomail.com	8281669345
9	43	Owen Owens	owenowens@yomail.com	8944701545
10	47	Randall Willis	randallwillis@yomail.com	6906132852
11	48	Cannon Jackson	cannonjackson@yomail.com	7427148917
12	49	Maxine Levine	maxinelevine@yomail.com	8782672355
13	457	Pacheco Pate	pachecopate@yomail.com	9062169671
14	58	Hillary Pennington	hillarypennington@yomail.com	7472166965
15	65	Carolina Morton	carolinamorton@yomail.com	8083398528
16	66	Dena Finley	denafinley@yomail.com	8352236222
17	68	Tanya Stark	tanyastark@yomail.com	6270566619

```
--MaxCases for a specific department, profession and resident cases
with q as ( SELECT p.id,p.resident_status as resident_status, trim(split_part(acad_status,',',1)) as profession,
                  trim(split_part(acad_status,',',2)) as department
            FROM person p
            RIGHT JOIN neo4j_results n
            ON p.name=n.personname)
SELECT department, profession, resident_status, COUNT(id) AS Count
from q
group by department,profession,resident_status
order by count desc~
```

Output Result 137 ×

department	profession	resident_status	count
1 Biological Sciences	Staff	Off campus	10
2 Biological Sciences	Masters	Off campus	9
3 Scripps Institution of Oceanography	Undergraduate	On campus	9
4 Physical Sciences	Masters	On campus	9
5 Data Science	PhD	On campus	9
6 Social Sciences	Undergraduate	Off campus	9
7 Data Science	Masters	Off campus	8
8 Scripps Institution of Oceanography	Masters	On campus	8
9 Health Sciences	PhD	Off campus	8
10 Social Sciences	Undergraduate	On campus	8
11 Physical Sciences	Staff	On campus	8
12 Engineering	Masters	Off campus	8
13 Social Sciences	Staff	Off campus	8
14 Data Science	Staff	On campus	7
15 Biological Sciences	Undergraduate	Off campus	7
16 Health Sciences	Professor	On campus	7
17 Engineering	Staff	On campus	7
18 Arts and Humanities	Staff	Off campus	7
19 Health Sciences	Staff	On campus	7

Query to find the count of affected people in a given department, profession and resident_status

Query to find positive people partitioned over their respective departments and profession

```
cur.execute("""WITH query2 AS ( SELECT p.name, trim(split_part(acad_status,',',1)) as profession,
    TRIM(split_part(acad_status,',',2)) as department
    FROM person p
    RIGHT JOIN neo4j_results n
    ON p.name=n.personname)
SELECT name,department,profession, COUNT(*) OVER(PARTITION BY department,profession) AS Count
FROM query2
GROUP BY department,profession,name
ORDER BY count desc
""")
```

```
cur.fetchall()
```

```
[('Emily Kramer', 'Social Sciences', 'Undergraduate', 17),
('Constance Santiago', 'Social Sciences', 'Undergraduate', 17),
('Barber Wilkins', 'Social Sciences', 'Undergraduate', 17),
('Boyle Wong', 'Social Sciences', 'Undergraduate', 17),
('Castaneda Dudley', 'Social Sciences', 'Undergraduate', 17),
('Ferrell Arnold', 'Social Sciences', 'Undergraduate', 17),
('Goff Barker', 'Social Sciences', 'Undergraduate', 17),
('Goff Pacheco', 'Social Sciences', 'Undergraduate', 17),
('Greer Lambert', 'Social Sciences', 'Undergraduate', 17),
('Karen Rosario', 'Social Sciences', 'Undergraduate', 17),
('Karin Estrada', 'Social Sciences', 'Undergraduate', 17),
('Pearlie Marquez', 'Social Sciences', 'Undergraduate', 17),
('Schultz Bright', 'Social Sciences', 'Undergraduate', 17),
('Vargas Sawyer', 'Social Sciences', 'Undergraduate', 17),
('Verna Nielsen', 'Social Sciences', 'Undergraduate', 17),
('Victoria Walton', 'Social Sciences', 'Undergraduate', 17),
('Young Glenn', 'Social Sciences', 'Undergraduate', 17),
('Beasley Mcfadden', 'Social Sciences', 'Staff', 14),
('Banks Sutton', 'Social Sciences', 'Staff', 14),
```

```
with query2 as ( SELECT p.name, trim(split_part(acad_status,',',1)) as profession,
    trim(split_part(acad_status,',',2)) as department
    FROM person p
    RIGHT JOIN neo4j_results n
    ON p.name=n.personname)
SELECT name,department,profession, COUNT(*) OVER(PARTITION BY department,profession) AS Count
FROM query2
GROUP BY department,profession,name
ORDER BY count desc
```

	name	department	profession	count
1	Emily Kramer	Social Sciences	Undergraduate	17
2	Constance Santiago	Social Sciences	Undergraduate	17
3	Barber Wilkins	Social Sciences	Undergraduate	17
4	Boyle Wong	Social Sciences	Undergraduate	17
5	Castaneda Dudley	Social Sciences	Undergraduate	17
6	Ferrell Arnold	Social Sciences	Undergraduate	17
7	Goff Barker	Social Sciences	Undergraduate	17
8	Goff Pacheco	Social Sciences	Undergraduate	17
9	Greer Lambert	Social Sciences	Undergraduate	17
10	Karen Rosario	Social Sciences	Undergraduate	17
11	Karin Estrada	Social Sciences	Undergraduate	17
12	Pearlie Marquez	Social Sciences	Undergraduate	17
13	Schultz Bright	Social Sciences	Undergraduate	17
14	Vargas Sawyer	Social Sciences	Undergraduate	17
15	Verna Nielsen	Social Sciences	Undergraduate	17
16	Victoria Walton	Social Sciences	Undergraduate	17
17	Young Glenn	Social Sciences	Undergraduate	17
18	Beasley Mcfadden	Social Sciences	Staff	14
19	Banks Sutton	Social Sciences	Staff	14
20	Baker Doyle	Social Sciences	Staff	14

Query to find the percentage of people living in vicinity of UCSD campus and tested COVID +ve

```
#Query to find the percentage of people who tested positive among the residents of the UCSD region
cur.execute("""WITH query3 AS (SELECT cast(COUNT(*) AS numeric) as count1
                           FROM person
                           WHERE addresslong between -117.248564 and -117.216691 and
                                 addesslat between 32.863066 and 32.899635),
query4 as (SELECT cast(COUNT(*) AS numeric) as count2
                           FROM person p
                           RIGHT JOIN neo4j_results n
                           ON p.name=n.personname
                           WHERE p.addresslong between -117.248564 and -117.216691 and
                                 p.addesslat between 32.863066 and 32.899635)
SELECT concat(round(((count2/count1)*100),2),'%') as PositiveCases_UCSD
FROM query3,query4
""")
```

```
cur.fetchall()
```

```
[('31.71%',)]
```

```
with query3 as (SELECT cast(COUNT(*) AS numeric) as count1
                           FROM person
                           WHERE addresslong between -117.248564 and -117.216691 and
                                 addesslat between 32.863066 and 32.899635),
query4 as (SELECT cast(COUNT(*) AS numeric) as count2
                           FROM person p
                           RIGHT JOIN neo4j_results n
                           ON p.name=n.personname
                           WHERE p.addresslong between -117.248564 and -117.216691 and
                                 p.addesslat between 32.863066 and 32.899635)
SELECT concat(round(((count2/count1)*100),2),'%') as PositiveCases_UCSD
FROM query3,query4
```

positivecases_ucsd	
1	31.71%

MongoDB & JsonPath

```
_id: ObjectId('6390de14a863bf2a85d978fe')
UniqueID: "63819d2b9e5932c0b18ab0f0"
index: 3
picture: "http://picSite.com/64x64"
name: "Hilda Henry"
address: "112 Irving Street, Albrightsville, Kansas, 3308"
latitude: 32.916609
longitude: -117.157861
Company: "JETSILK"
✓ FamilyMembers: Array
  ✓ 0: Object
    id: 1
    name: "Harrell Cain"
    age: 16
    Relationship: "Child"
    isStayingAlong: true
  > 1: Object
  ✓ Colleagues: Array
    ✓ 0: Object
      name: "Weeks Day"
      CompanyID: 4409759
      email: "weeksday@jetsilk.com"
      age: 25
    > 1: Object
      name: "Hutchinson Hester"
      CompanyID: 9569579
      email: "hutchinsonhester@jetsilk.com"
      age: 52
    > 2: Object
      name: "Allison Hull"
      CompanyID: 9657508
      email: "allisonhull@jetsilk.com"
      age: 33
    > 3: Object
      name: "Blackwell Boyle"
      CompanyID: 5166745
      email: "blackwellboyle@jetsilk.com"
```

Details of positive people

```
In [123]: #MongoDB - Details of people tested positive from MongoDB
```

```
listnew=[]
for res in entire_result:
    mydoc = collection_1.find({"name": res})
    for x in mydoc:
        listnew.append(x)
print(listnew)
```

```
[{"_id": ObjectId('6390de14a863bf2a85d97904'), 'UniqueID': '63819d2b68daaa912fa6b86d', 'index': 9, 'picture': 'http://picSite.com/64x64', 'name': 'Decker Warner', 'address': '437 Vanderveer Street, Greenfields, Virginia, 7449', 'latitude': 32.690984, 'longitude': -117.132488, 'Company': 'MIXERS', 'FamilyMembers': [{"id": 1, "name": "Ayers James", "age": 52, "Relationship": "Sibling", "isStayingAlong": False}, {"id": 2, "name": "Heath Steele", "age": 25, "Relationship": "Friend & Others", "isStayingAlong": True}, {"id": 3, "name": "Howard Holland", "age": 58, "Relationship": "Parent", "isStayingAlong": False}], 'Colleagues': [{"name": "Hawkins Walker", "CompanyID": 1643983, "email": "hawkinswalker@mixers.com", "age": 33}, {"name": "Raymond Hardin", "CompanyID": 6677594, "email": "raymondhardin@mixers.com", "age": 52}, {"name": "Noreen Garcia", "CompanyID": 5466037, "email": "nreengarcia@mixers.com", "age": 47}]}], {"_id": ObjectId('6390de14a863bf2a85d97908'), 'UniqueID': '63819d2bd223f7b99bdfe798', 'index': 13, 'picture': 'http://picSite.com/64x64', 'name': 'Salazar Hicks', 'address': '686 Seton Place, Norris, South Carolina, 5592', 'latitude': 32.879848, 'longitude': -117.12107, 'Company': 'HANDSHAKE', 'FamilyMembers': [{"id": 1, "name": "Wilkerson Good", "age": 66, "Relationship": "Child", "isStayingAlong": False}, {"id": 2, "name": "Bean Powers", "age": 61, "Relationship": "Grandparent", "isStayingAlong": False}, {"id": 3, "name": "Estelle Barnes", "age": 3, "Relationship": "Parent", "isStayingAlong": False}, {"id": 4, "name": "Ross Padilla", "age": 15, "Relationship": "Child", "isStayingAlong": False}], 'Colleagues': [{"name": "Frost Flores", "CompanyID": 9519731, "email": "frostflores@handshake.com", "age": 28}, {"name": "Kari Adams", "CompanyID": 6883109, "email": "kariadams@handshake.com", "age": 44}, {"name": "Genevieve Leonard", "CompanyID": 5291514, "email": "genevieveleonard@handshake.com", "age": 40}]}], {"_id": ObjectId('6390de14a863bf2a85d97909'), 'UniqueID': '63819d2bb45b46d5682bd945', 'index': 14, 'picture': 'http://picSite.com/64x64', 'name': 'Wong Malone', 'address': '548 Falmouth Street, Volta, American Samoa, 3249', 'latitude': 32.744215, 'longitude': -117.133276, 'notes': "I'm not sure if I can get into this place because I don't have a job yet."}]
```

Retrieving the details of family members living along with the people who tested COVID +ve:

```
In [124]: listnew=[]
for res in entire_result:
    mydoc = collection_1.find({"name": res, "FamilyMembers.isStayingAlong":True},
                               {"FamilyMembers.name":1,"name":1,"FamilyMembers.Relationship":1,"address":1})
    for x in mydoc:
        listnew.append(x)

print(listnew)
```

[{'_id': ObjectId('6390de14a863bf2a85d97904'), 'name': 'Decker Warner', 'address': '437 Vanderveer Street, Greenfield s, Virginia, 7449', 'FamilyMembers': [{ 'name': 'Ayers James', 'Relationship': 'Sibling'}, { 'name': 'Heath Steele', 'Relationship': 'Friend & Others'}, { 'name': 'Howard Holland', 'Relationship': 'Parent'}]}, {'_id': ObjectId('6390de14a863bf2a85d97909'), 'name': 'Wong Malone', 'address': '548 Falmouth Street, Volta, American Samoa, 3249', 'FamilyMembers': [{ 'name': 'Maryanne Pugh', 'Relationship': 'Friend & Others'}, { 'name': 'Joni Rodgers', 'Relationship': 'Child'}, { 'name': 'Elliott Keller', 'Relationship': 'Grandchild'}]}, {'_id': ObjectId('6390de14a863bf2a85d97915'), 'name': 'Velez Mcclure', 'address': '509 Midwood Street, Centerville, Michigan, 7344', 'FamilyMembers': [{ 'name': 'Becker Lee', 'Relationship': 'Grandchild'}, { 'name': 'Rutledge Henson', 'Relationship': 'Parent'}, { 'name': 'Wynn Holt', 'Relationship': 'Sibling'}, { 'name': 'Flora Burke', 'Relationship': 'Friend & Others'}, { 'name': 'Rios Hampton', 'Relationship': 'Grandparent'}]}, {'_id': ObjectId('6390de14a863bf2a85d97917'), 'name': 'Robbins Koch', 'address': '872 Devoe Street, Grazierville, Tennessee, 1824', 'FamilyMembers': [{ 'name': 'Cruz Stevens', 'Relationship': 'Sibling'}, { 'name': 'Ayala Gross', 'Relationship': 'Parent'}, { 'name': 'Penelope Barrera', 'Relationship': 'Sibling'}, { 'name': 'Ran dolph Guerrero', 'Relationship': 'Sibling'}]}, {'_id': ObjectId('6390de14a863bf2a85d97920'), 'name': 'Mildred Mullins', 'address': '137 Scholes Street, Murillo, District Of Columbia, 125', 'FamilyMembers': [{ 'name': 'Vonda Calderon', 'Relationship': 'Friend & Others'}, { 'name': 'Staci Becker', 'Relationship': 'Grandparent'}, { 'name': 'Yesenia Wals h', 'Relationship': 'Grandchild'}, { 'name': 'Malinda Ward', 'Relationship': 'Friend & Others'}]}, {'_id': ObjectId('6390de14a863bf2a85d97923'), 'name': 'Jenifer White', 'address': '814 Colonial Court, Bluffview, Federated States Of Micronesia, 7035', 'FamilyMembers': [{ 'name': 'Montoya Dyer', 'Relationship': 'Spouse'}, { 'name': 'Dolly Woodard', 'Relationship': 'Child'}, { 'name': 'Washington Powell', 'Relationship': 'Child'}, { 'name': 'Marci Church', 'Relationship': 'Child'}]}

Visualising the retrieved family members' details using a dataframe object:

```
listnew=[ ]
for res in entire_result:
    myquery = {"name": res, "FamilyMembers.isStayingAlong":True}
    mydoc = collection_1.find(myquery,{"FamilyMembers.name":1,"name":1,"FamilyMembers.Relationship":1,"address":1})
    for x in mydoc:
        listnew.append(x)
        json_data = x
        json_exp=parse('$.FamilyMembers')
        for match in json_exp.find(json_data):
            countmemb=len(match.value)
            names.append(x[ 'name' ])
            count_family_along.append(countmemb)
            family_members.append(match.value)
            addresses.append(x[ "address" ])
```

		name	count_family	family_members	address
92	Lorie Cummings	10	[{"name": "Battle Richardson", "Relationship": "S..."}, {"name": "Lorie Cummings", "Relationship": "S..."}, {"name": "Cassidy Williams", "Relationship": "S..."}, {"name": "Dawn Jackson", "Relationship": "S..."}, {"name": "Kathy..."}, {"name": "Lorie Cummings", "Relationship": "S..."}, {"name": "Lorie Cummings", "Relationship": "S..."}]	520 Jerome Avenue, Guilford, Arkansas, 2814	
30	Garcia Rutledge	10	[{"name": "Bettie Rosales", "Relationship": "S..."}, {"name": "Garcia Rutledge", "Relationship": "S..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	286 Alice Court, Waiohinu, Federated States Of...	
235	Emily Kramer	10	[{"name": "Cherry Patel", "Relationship": "Sib..."}, {"name": "Emily Kramer", "Relationship": "Sib..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	494 Coleridge Street, Noxen, Connecticut, 4622	
194	Francesca Rush	10	[{"name": "Mccarthy Sandoval", "Relationship": "S..."}, {"name": "Francesca Rush", "Relationship": "S..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	675 Ralph Avenue, Macdona, Rhode Island, 8345	
203	Rollins Pittman	10	[{"name": "Adkins Reeves", "Relationship": "Ch..."}, {"name": "Rollins Pittman", "Relationship": "Ch..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	806 Tennis Court, Tioga, Connecticut, 2388	
207	Britt Stone	10	[{"name": "Ochoa Mcdonald", "Relationship": "S..."}, {"name": "Britt Stone", "Relationship": "S..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	450 Harbor Lane, Iola, Colorado, 5407	
214	Ewing Thomas	10	[{"name": "Craft Pope", "Relationship": "Grand..."}, {"name": "Ewing Thomas", "Relationship": "Grand..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	528 Cooper Street, Chloride, Arkansas, 1672	
282	Vargas Sawyer	9	[{"name": "Forbes Jensen", "Relationship": "Si..."}, {"name": "Vargas Sawyer", "Relationship": "Si..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	992 Kosciusko Street, Temperanceville, Guam, 3291	
239	Maribel Arnold	9	[{"name": "Carmen Mckinney", "Relationship": "..."}, {"name": "Maribel Arnold", "Relationship": "..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	975 Duffield Street, Hilltop, Michigan, 8290	
134	Alisa Kramer	9	[{"name": "Ruthie Patel", "Relationship": "Spo..."}, {"name": "Alisa Kramer", "Relationship": "Spo..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	181 Graham Avenue, Greenbackville, Wyoming, 5931	
310	Vanessa Hewitt	9	[{"name": "Hensley Faulkner", "Relationship": "..."}, {"name": "Vanessa Hewitt", "Relationship": "..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	974 Hope Street, Blackgum, Maine, 3116	
243	Nolan Eaton	9	[{"name": "Tamika Martinez", "Relationship": "..."}, {"name": "Nolan Eaton", "Relationship": "..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	600 Calyer Street, Aurora, Idaho, 2435	
41	Patrice Floyd	9	[{"name": "Janette Short", "Relationship": "Gr..."}, {"name": "Patrice Floyd", "Relationship": "Gr..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	423 Ocean Avenue, Romeville, Rhode Island, 4048	
220	Young Glenn	9	[{"name": "Spence Moore", "Relationship": "Gra..."}, {"name": "Young Glenn", "Relationship": "Gra..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	746 Irving Street, Albrightsville, New Hampshire...	
85	White Dean	9	[{"name": "Conway Bass", "Relationship": "Sib..."}, {"name": "White Dean", "Relationship": "Sib..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	444 Engert Avenue, Washington, Virgin Islands,...	
54	Natalie Robbins	8	[{"name": "Dominique Estes", "Relationship": "..."}, {"name": "Natalie Robbins", "Relationship": "..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	895 Eaton Court, Saranap, District Of Columbia...	
287	Lou Case	8	[{"name": "Mercer Donovan", "Relationship": "F..."}, {"name": "Lou Case", "Relationship": "F..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	730 Bay Avenue, Accoville, Virgin Islands, 8210	
260	Kirsten Doyle	8	[{"name": "Pearl Valencia", "Relationship": "G..."}, {"name": "Kirsten Doyle", "Relationship": "G..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	232 Roebling Street, Riverton, Utah, 6983	
225	Clarissa Burton	8	[{"name": "Barrett Velazquez", "Relationship": "..."}, {"name": "Clarissa Burton", "Relationship": "..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	735 Sullivan Street, Dola, Indiana, 2385	
256	Curry Humphrey	8	[{"name": "Case Haley", "Relationship": "Sib..."}, {"name": "Curry Humphrey", "Relationship": "Sib..."}, {"name": "Lorraine..."}, {"name": "Lorraine..."}]	618 Ocean Avenue, Romeville, North Dakota, 3533	

Retrieving people affected in a particular list of companies:

```
companydata = []
for company in companylist:
    for res in entire_result:
        mydoc = collection_1.find({"name":res,"Company":company},
                                    {"name":1,"Company":1,"Colleagues":1,"_id":0})
        for i in mydoc:
            companydata.append(i)

print(companydata)
```

```
[{'name': 'Lynda McCormick', 'Company': 'RENOVIZE', 'Colleagues': [{'name': 'Earnestine Boyer', 'CompanyID': 8314794, 'email': 'earnestineboyer@renovize.com', 'age': 50}, {'name': 'Dionne Duffy', 'CompanyID': 7049209, 'email': 'dionneduffy@renovize.com', 'age': 56}, {'name': 'Melody Allen', 'CompanyID': 4921244, 'email': 'melodyallen@renovize.com', 'age': 56}, {'name': 'Campbell Stein', 'CompanyID': 3770740, 'email': 'campbellstein@renovize.com', 'age': 57}], {'name': 'Francis Owens', 'Company': 'NURPLEX', 'Colleagues': [{'name': 'Christa Pope', 'CompanyID': 4469043, 'email': 'christapope@nurplex.com', 'age': 44}, {'name': 'Helena Blackwell', 'CompanyID': 2317993, 'email': 'helenablackwell@nurplex.com', 'age': 33}], {'name': 'Bowen Mcfarland', 'Company': 'PULZE', 'Colleagues': [{'name': 'Patterson Sharp', 'CompanyID': 4133477, 'email': 'pattersonsharp@pulze.com', 'age': 21}, {'name': 'Arnold Rios', 'CompanyID': 6954286, 'email': 'arnoldrios@pulze.com', 'age': 56}, {'name': 'Noble Kramer', 'CompanyID': 4069339, 'email': 'noblekramer@pulze.com', 'age': 42}, {'name': 'Eliza Patel', 'CompanyID': 8636887, 'email': 'elizapatel@pulze.com', 'age': 33}, {'name': 'Guadalupe McLaughlin', 'CompanyID': 9783454, 'email': 'guadalupemclaughlin@pulze.com', 'age': 43}, {'name': 'Halley Sloan', 'CompanyID': 7328827, 'email': 'haleysloan@pulze.com', 'age': 52}], {'name': 'George Campos', 'Company': 'MUSAPHICS', 'Colleagues': [{'name': 'Gillespie Sellers', 'CompanyID': 8741550, 'email': 'gillespiesellers@musaphics.com', 'age': 49}]}, {'name': 'White Dean', 'Company': 'ACCUPHARM', 'Colleagues': [{'name': 'Woodward Reyes', 'CompanyID': 1201262, 'email': 'woodwardreyes@accupharm.com', 'age': 43}, {'name': 'Alexis Arnold', 'CompanyID': 9566280, 'email': 'alexisarnold@accupharm.com', 'age': 29}, {'name': 'Lorraine McKinney', 'CompanyID': 4491405, 'email': 'lorrainenmckinney@accupharm.com', 'age': 22}, {'name': 'Serrano Robinson', 'CompanyID': 1182038, 'email': 'serranorobinson@accupharm.com', 'age': 59}, {'name': 'Angeline Fletcher', 'CompanyID': 7490122, 'email': 'angelinefletcher@accupharm.com', 'age': 44}, {'name': 'Mcneil Grimes', 'CompanyID': 3945878, 'email': 'mcneilgrimes@accupharm.com', 'age': 47}, {'name': 'Mitchell Garrett', 'CompanyID': 9496454, 'email': 'mitchellgarrett@accupharm.com', 'age': 59}], {'name': 'Karen Rosario', 'Company': 'GEEKKO', 'Colleagues': [{'name': 'April Shannon', 'CompanyID': 9534423, 'email': 'aprilsannon@geekko.com', 'age': 52}, {'name': 'Berger Stewart', 'CompanyID': 4302492, 'email': 'bergerstewart@geekko.com', 'age': 28}, {'name': 'Short Levine', 'CompanyID': 8668721, 'email': 'shortlevine@geekko.com', 'age': 29}, {'name': 'Hamilton Frederick', 'CompanyID': 8849703, 'email': 'hamiltonfrederick@geekko.com', 'age': 51}]]
```

```

companydata=[]
for company in companylist:
    for res in entire_result:
        mydoc = collection_1.find({"name":res,"Company":company},
                                  {"name":1,"Company":1,"Colleagues":1,"_id":0})
        for i in mydoc:
            companydata.append(i)
            json_exp=parse('$.Colleagues')
            json_expl=parse('$.name')
            json_exp2=parse('$.Company')
            for match in json_exp2.find(i):
                print("Company Name:", match.value)
                company_name.append(match.value)
            for match in json_expl.find(i):
                print(match.value)
                employees_affected.append(match.value)
            for match in json_exp.find(i):
                for j in match.value:
                    print(j['name'])
                    employees_affected.append(j['name'])
print("-----")

```

Company Name: RENOVIZE
 Lynda McCormick
 Earnestine Boyer
 Dionne Duffy
 Melody Allen
 Campbell Stein

 Company Name: NURPLEX
 Francis Owens
 Christa Pope
 Helena Blackwell

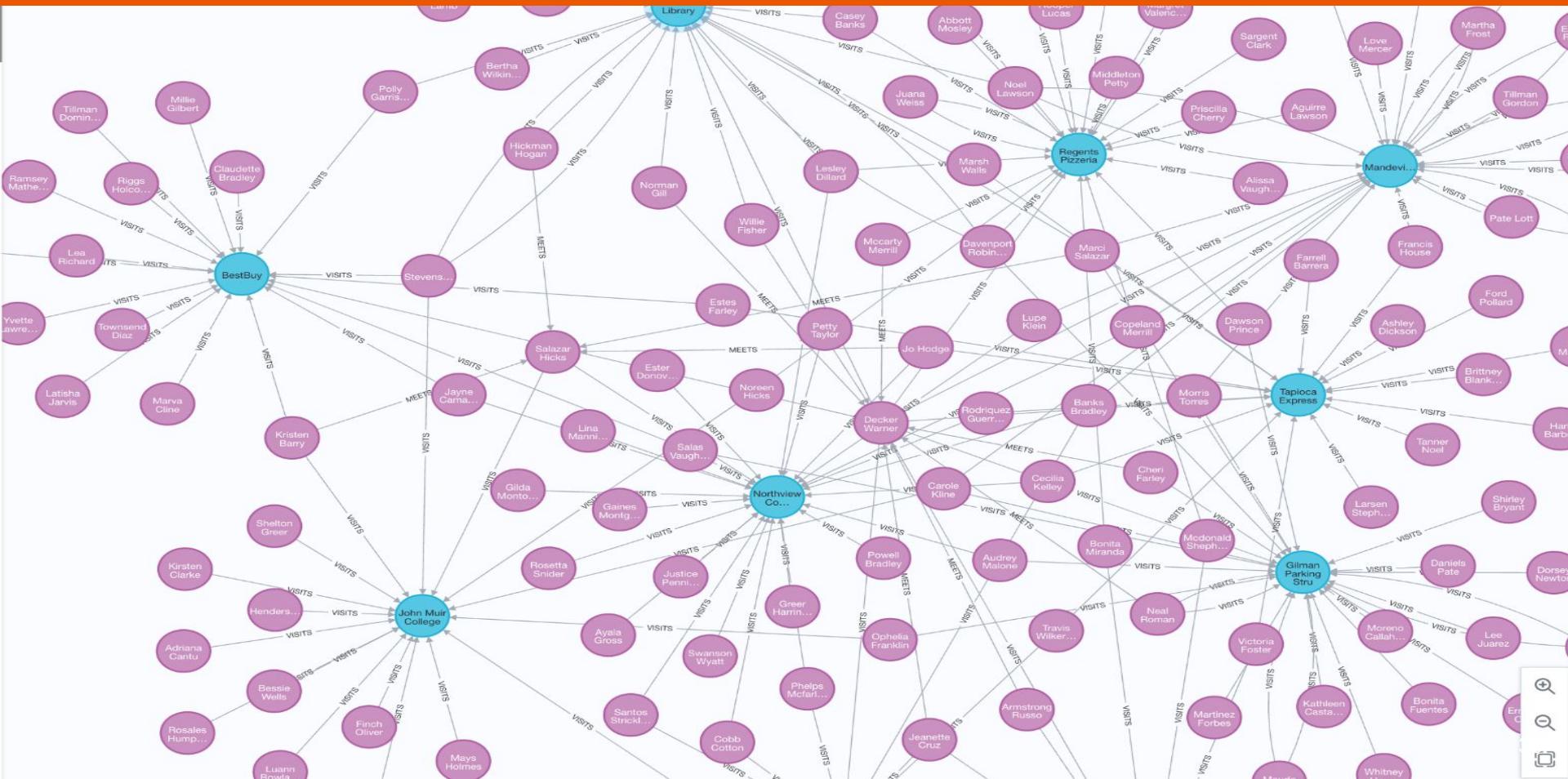
 Company Name: PULZE
 Bowen Mcfarland
 Patterson Sharp
 Arnold Rios
 Noble Kramer
 Eliza Patel
 Guadalupe McLaughlin
 Haley Sloan

 Company Name: MUSAPHICS
 George Campos
 Gillespie Sellers

 Company Name: ACCUPHARM
 White Dean
 Woodward Reyes
 Alexis Arnold
 Lorraine McKinney
 Serrano Robinson
 Angeline Fletcher
 Mcneil Grimes
 Mitchell Garrett

 Company Name: GEEKKO
 Karen Rosario
 April Shannon
 Berger Stewart
 Short Levine
 Hamilton Frederick

Neo4j



Creating a MEETS relationship:

```
1 match (npp:Results)-[vis1:VISITS]-(pl:Places)←[vis2:VISITS]-(pp:Results{covidStatus:"Positive"})
2 WHERE npp.covidStatus⊸ "Positive"
3 with npp, pp, CASE WHEN vis1.entryTime.epochMillis>vis2.entryTime.epochMillis THEN vis1.entryTime.epochMillis ELSE
4 vis2.entryTime.epochMillis END as maxStart, CASE WHEN vis1.exitTime.epochMillis<vis2.exitTime.epochMillis THEN
5 vis1.exitTime.epochMillis ELSE vis2.exitTime.epochMillis END as minEnd
6 WHERE maxStart≤minEnd
7 with npp, pp, sum(minEnd-maxStart)*(0.001/3600) as meetTime
8 create (npp)-[:MEETS{meettime: duration({minutes:meetTime})}]->(pp);
```

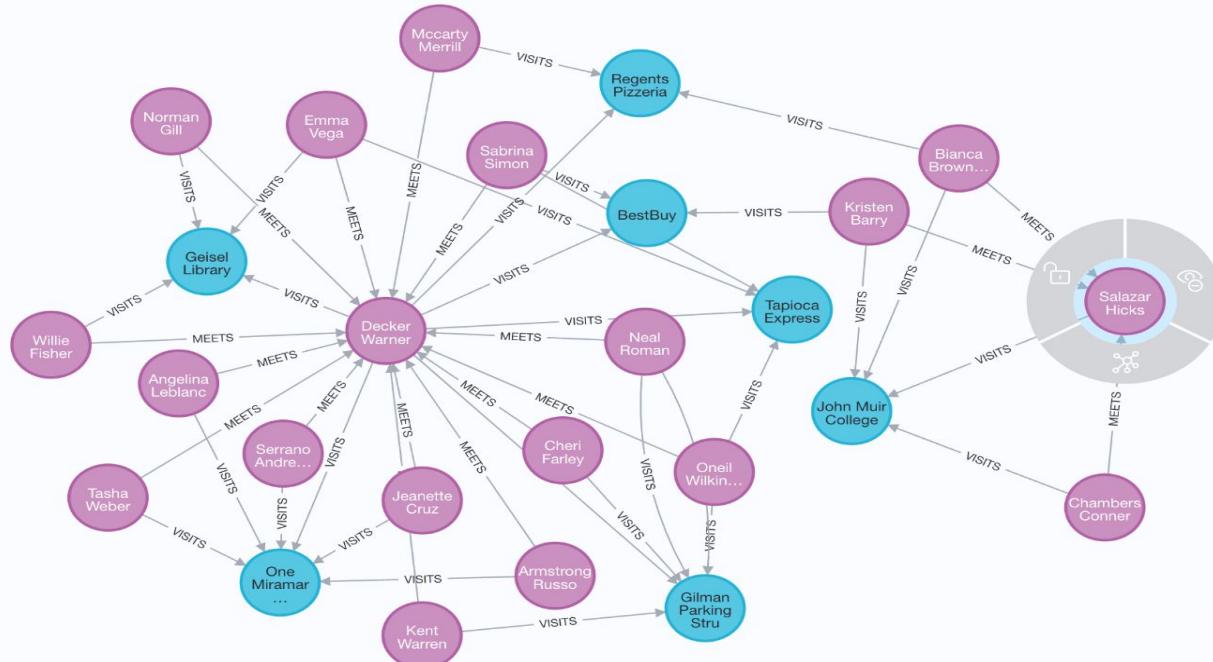
Set 4167 properties, created 4167 relationships, completed after 1372 ms.



Connection between COVID affected person and not affected person through the common places they visit:

```
1 match (p:Results{covidStatus:"Positive"})-[v1:VISITS]→(pl:Places)
2 with p,v1,pl
3 limit 10
4 match path = (p)-[v1]→(pl)←[v2:VISITS]-(p2:Results)
5 with p2,path, CASE WHEN v1.entryTime.epochMillis>v2.entryTime.epochMillis THEN v1.entryTime.epochMillis ELSE
v2.entryTime.epochMillis END as maxStart, CASE WHEN v1.exitTime.epochMillis<v2.exitTime.epochMillis THEN v1.exitTime.epochMillis
ELSE v2.exitTime.epochMillis END as minEnd
6 WHERE maxStart<minEnd and p2.covidStatus≠"Positive"
7 return path;
8
```

Graph
Table
Text
Code



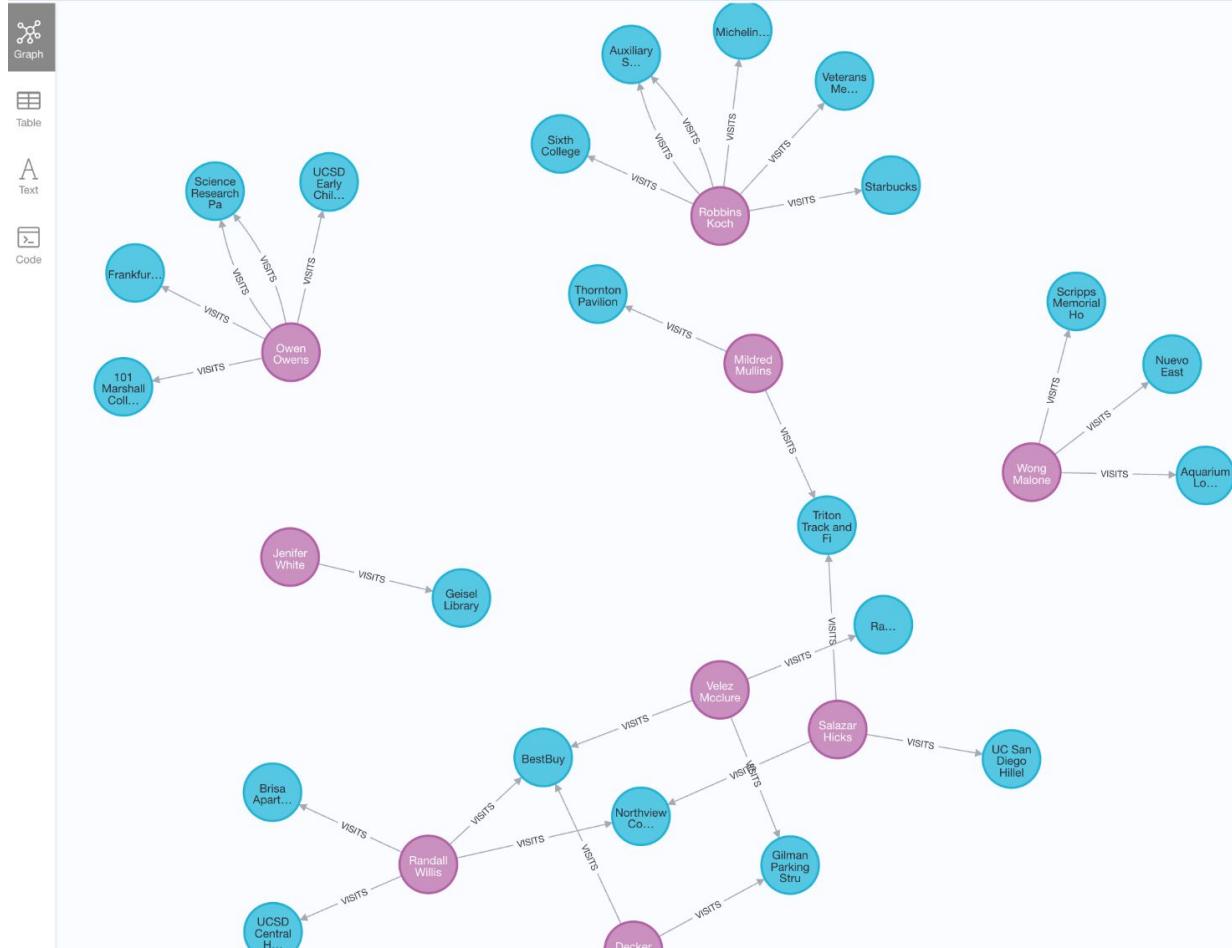
Node properties

Results

<id> 3330
confirmedT "2022-11-17T00:52:52Z"
ime
covidStatus Positive
personId 13
personNam Salazar Hicks
e

Places visited by people after being tested COVID +ve

```
1 match (p:Results{covidStatus:"Positive"})-[v1:VISITS]→(pl:Places)
2 where p.confirmedTime < v1.entryTime
3 return p, v1, pl
4 limit 25
```



Overview

Node labels

* (33) Results (9) Places (24)

Relationship types

* (31) VISITS (31)

Displaying 33 nodes, 25 relationships.

```
1 match (p:Results{covidStatus:"Positive"})-[v1:VISITS]→(pl:Places)
2 where p.confirmedTime < v1.entryTime
3 return p.personId, count(pl) as countPlacesVisited
4 order by count(pl) desc
5 limit 50;
6
```

Table

A
Text

Code

	p.personId	countPlacesVisited
1	"169"	12
2	"1164"	9
3	"1267"	9
4	"317"	9
5	"1315"	9
6	"595"	9
7	"792"	8

**People more susceptible to
COVID - Based on overlap time
of visits with COVID +ve people**

```

1 match (npp:Results)-[vis1:VISITS]-(pl:Places)←[vis2:VISITS]-(pp:Results{covidStatus:"Positive"})
2 WHERE npp.covidStatus $\neq$ "Positive"
3 with npp, CASE WHEN vis1.entryTime.epochMillis>vis2.entryTime.epochMillis THEN vis1.entryTime.epochMillis ELSE
vis2.entryTime.epochMillis END as maxStart, CASE WHEN vis1.exitTime.epochMillis<vis2.exitTime.epochMillis THEN
vis1.exitTime.epochMillis ELSE vis2.exitTime.epochMillis END as minEnd
4 WHERE maxStart $\leq$ minEnd
5 return npp.personName, npp.covidStatus, sum(minEnd-maxStart)*(0.001/3600) as overlaptIME
6 order by overlaptIME desc
7 limit 100;

```

Table

Text

Code

	npp.personName	npp.covidStatus	overlaptIME
1	"Aguirre Lawson"	"Negative"	101.81416666666667
2	"Marci Salazar"	"Negative"	101.0216666666666
3	"Calderon Gibson"	"Negative"	98.61861111111111
4	"Maxine Kim"	"Inconclusive"	96.90833333333333
5	"Baldwin Franklin"	"Inconclusive"	96.20333333333333
6	"Klein Mooney"	"Negative"	93.13333333333333
7	"Ana Gamble"	"Inconclusive"	89.36222222222221
8	"Burton Bradshaw"	"Negative"	88.68249999999999
9	"Jessie Chapman"	"Negative"	85.46777777777777

Spotting COVID hotspots

```

1 match (p:Results {covidStatus: "Positive"})-[vis:VISITS]→(pl:Places)
2 with p, pl.placeName as placename, count(vis) as nrofsickvisits
3 match path = (p)-[vis2:VISITS]→(pl:Places)
4 with pl.placeName as placename, nrofsickvisits as nrofsickvisits, count(vis2) as totalnrofvisits
5 limit 1000
6 return placename, nrofsickvisits, totalnrofvisits, roundtoFloat(nrofsickvisits)/toFloat(totalnrofvisits)*10000/100 as
percentageofsickvisits;
7

```

Table

	placename	nrofsickvisits	totalnrofvisits	percentageofsickvisits
151	"One Miramar Street Apartments"	2	7	28.57
152	"Brisa Apartment"	2	4	50.0
153	"Rita Atkinson Residences"	2	8	25.0
154	"Le Natalie OD"	2	15	13.33
155	"UCSD Central Mesa Houses"	2	3	66.67
156	"Scripps Institution of Oceanography"	2	8	25.0
157	"Matthew W. Gentile, OD"	2	7	28.57
158	"Porton Nuevo East"	2	2	100.0
159	"Price Center Theater"	2	10	20.0

Conclusion

CONCLUSION:

A lot of use cases were discussed and analysed and there is more scope in "Contact tracing".

- Analysed the susceptibility of COVID in each department, profession and resident status of UCSD.
- Analysed the spread of COVID cases in the vicinity of UCSD using the longitudes and latitudes.
- Retrieved information about the family members who are staying along with the COVID infected people.
- Retrieved susceptible colleagues of COVID infected person who belonged to a particular Company.
- People who went to the same place as that of the infected person at the same time duration were found to be susceptible.
- Places visited by people after being tested positive for covid was found.
- COVID hotspots were detected.

THANK YOU
