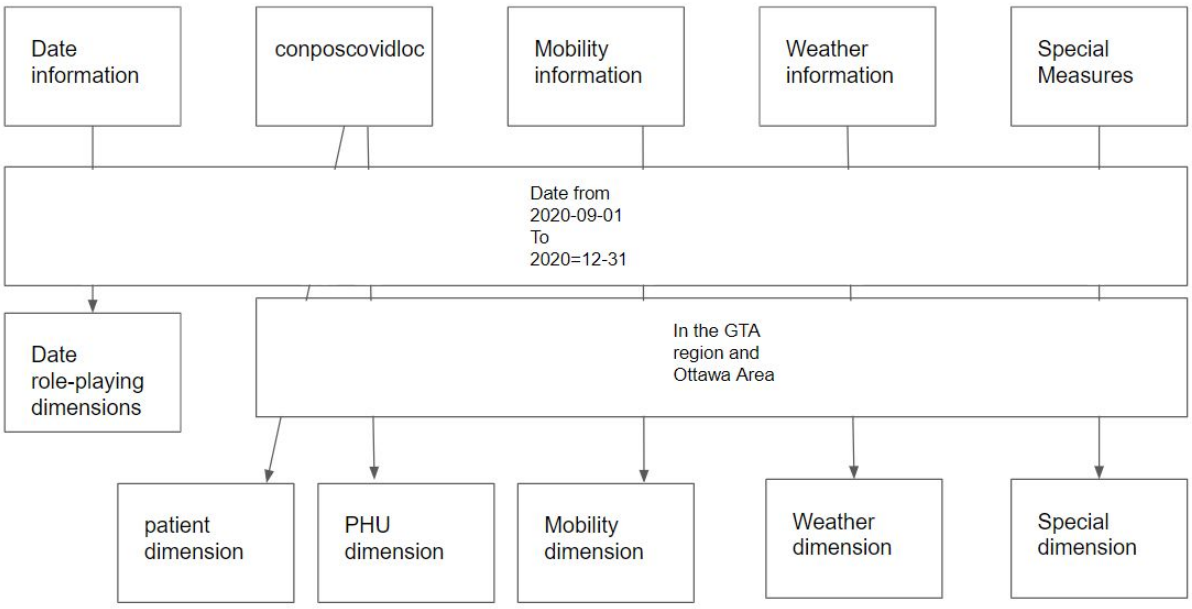
Physical Design and Data Staging

Group：18
Chuhao Jia
Xiaohan Yu
Yiwen Liu

In this report, we will discuss how we finish the project, the problem that we meet in the project and how we deal with it.

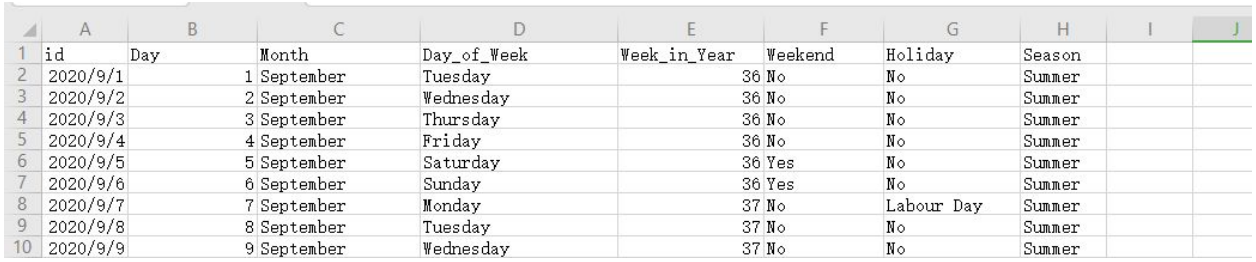
BTW: The csv sample shown in the report may not be the same as in the final csv we submit because they are screened during the testing.

0:high-level data staging plan



1: Staging of Date dimension

In this part, I know all of these 4 date dimensions should save the same data. So I First search the daily data from 2020-09-01 to 2020-12-31, which is the target period for our group. Then I create a date.csv to save the data.



	A	B	C	D	E	F	G	H	I	J
1	id	Day	Month	Day_of_Week	Week_in_Year	Weekend	Holiday	Season		
2	2020/9/1	1	September	Tuesday	36	No	No	Summer		
3	2020/9/2	2	September	Wednesday	36	No	No	Summer		
4	2020/9/3	3	September	Thursday	36	No	No	Summer		
5	2020/9/4	4	September	Friday	36	No	No	Summer		
6	2020/9/5	5	September	Saturday	36	Yes	No	Summer		
7	2020/9/6	6	September	Sunday	36	Yes	No	Summer		
8	2020/9/7	7	September	Monday	37	No	Labour Day	Summer		
9	2020/9/8	8	September	Tuesday	37	No	No	Summer		
10	2020/9/9	9	September	Wednesday	37	No	No	Summer		

(sample)

Then we use python, import pandas to read date.csv and clone it 4 times to get 4 role-playing date dimensions. Of course we need to set the key and surrogate key for it, so do the numerical optimization like changing the date type from yyyy/mm/dd to yyyy-mm-dd for unifying format.

```
date = pd.read_csv("date.csv")

# print(date.head())

# create Onset Date dimension

Onset_date_dimension = pd.DataFrame(
    columns=['Onset_date_key', 'Date', 'Day', 'Month', 'Day_of_Week',
             'Week_in_Year', 'Weekend', 'Holiday', 'Season'])

for idx, row in date.iterrows():
    date_row = ["On"+row["id"].replace("-", ""), row["id"], row['Day'], row['Month'], row['Day_of_Week'],
                row['Week_in_Year'], row['Weekend'], row['Holiday'], row['Season']]
    Onset_date_dimension.loc[len(Onset_date_dimension)] = date_row

Onset_date_dimension.insert(0, "Onset_date_surrogate_key", np.arange(len(Onset_date_dimension)))
Onset_date_dimension.to_csv("Onset_Date_dimension.csv", index=False)
```

	A	B	C	D	E	F	G	H	I	J	K
1	Onset_date_surrogate_key	Onset_date_key	Date	Day	Month	Day_of_Week	Week_in_Y	Weekend	Holiday	Season	
2	0	On20200901	2020/9/1	1	September	Tuesday	36	No	No	Summer	
3	1	On20200902	2020/9/2	2	September	Wednesday	36	No	No	Summer	
4	2	On20200903	2020/9/3	3	September	Thursday	36	No	No	Summer	
5	3	On20200904	2020/9/4	4	September	Friday	36	No	No	Summer	
6	4	On20200905	2020/9/5	5	September	Saturday	36	Yes	No	Summer	
7	5	On20200906	2020/9/6	6	September	Sunday	36	Yes	No	Summer	
8	6	On20200907	2020/9/7	7	September	Monday	37	No	Labour Da	Summer	
9	7	On20200908	2020/9/8	8	September	Tuesday	37	No	No	Summer	
10	8	On20200909	2020/9/9	9	September	Wednesday	37	No	No	Summer	
11	9	On20200910	2020/9/10	10	September	Thursday	37	No	No	Summer	

(sample for one of the role-playing dimensions)

2: Staging of patient dimension

In this part, I notice that the conposcovidloc data is too large for testing the code, so I first create a new date to just keep about 50 samples in the original csv. We will also add data into it when we find that the samples we have do not match the information we need to retrieve. (Like the samples we have do not contain the York region, but we want to know if our code is working on York keyword)

Then we follow the steps showing in the giving PPT for our project to filter the required information for patient dimension. For the age group attribute, I am confused about some data like “<20”, because the type of these are different from the part like “50s”. But I don’t know how we would use the age group data in future, so I am not sure if I should keep these data as string type or just change it into int type, therefore I decide just keep them as the original csv do for later announcement

about the project. Also we add the surrogate key and patient key for it.

```
Patient_dimension = pd.DataFrame(columns=['Patient_Key', 'Gender', 'Age_group',
                                          'Acquisition_group', 'Outbreak_related'])
count = 0
for idx, row in df.iterrows():

    if int(row['Accurate_Episode_Date'][0:4]) == 2020 \
            and int(row['Accurate_Episode_Date'][5:7]) >= 9 \
            and int(row['Case_Reported_Date'][0:4]) == 2020 \
            and (row['Reporting_PHU'] == "Ottawa Public Health" or
                 row['Reporting_PHU'] == "Toronto Public Health" or
                 row['Reporting_PHU'] == "Durham Region Health Department" or
                 row['Reporting_PHU'] == "Halton Region Health Department" or
                 row['Reporting_PHU'] == "Peel Public Health" or
                 row['Reporting_PHU'] == "York Region Public Health Services"):

        if row['Outbreak_Related'] != "Yes":
            row['Outbreak_Related'] = "No"

        patient_row = [row["_id"], row['Client_Gender'], row['Age_Group'],
                       row['Case_AcquisitionInfo'], row['Outbreak_Related']]
        Patient_dimension.loc[len(Patient_dimension)] = patient_row

        count += 1
        print("current at ", idx)
        print(count, "added")

Patient_dimension.insert(0, "Patient_surrogate_key", np.arange(len(Patient_dimension)))
Patient_dimension.to_csv("patient_dimension.csv", index=False)
# print(Patient_dimension.head())
```

	A	B	C	D	E	F
1	Patient_surrogate_key	Patient_Key	Gender	Age_group	Acquisition_group	Outbreak_related
2	0	8000	MALE	70s	NO KNOWN EPI LINK	No
3	1	21003	MALE	60s	TRAVEL	No
4	2	21958	FEMALE	20s	CC	No
5	3	21961	MALE	20s	CC	No
6	4	21963	MALE	50s	CC	No
7	5	21964	MALE	20s	CC	No
8	6	21965	MALE	<20	CC	No
9	7	21966	MALE	30s	NO KNOWN EPI LINK	No
10	8	21967	MALE	20s	CC	No
11	9	21970	FEMALE	20s	CC	No
12	10	21974	MALE	<20	CC	No
13	11	21975	FEMALE	30s	CC	No
14	12	21976	FEMALE	60s	CC	No
15	13	21977	MALE	40s	NO KNOWN EPI LINK	No
16	14	21981	MALE	40s	CC	No
17	15	21983	FEMALE	20s	OB	Yes

3: Staging of PHU dimension

This part is funny. At the beginning, I save every data for every patient, but later I notice there are only about 30+ PHU exist. And I don’t have to record the repeated PHU data.

```python
# create PHU dimension

PHU_Location_dimension = pd.DataFrame(columns=['PHU_id', 'PHU_name', 'Address', 'City',
                                               'Postal_Code', 'Province', 'URL', 'Latitude', 'Longitude'])
pid = []

for idx, row in df.iterrows():
    if row['Outbreak_Related'] != "Yes":
        row['Outbreak_Related'] = "No"

    if row["Reporting_PHU_ID"] not in pid:
        pid.append(row["Reporting_PHU_ID"])

        PHU_Location_row = [row["Reporting_PHU_ID"], row['Reporting_PHU'], row['Reporting_PHU_Address'],
                            row['Reporting_PHU_City'], row['Reporting_PHU_Postal_Code'], "ON",
                            row['Reporting_PHU_Website'], row['Reporting_PHU_Latitude'], row['Reporting_PHU_Longitude']]
        PHU_Location_dimension.loc[len(PHU_Location_dimension)] = PHU_Location_row

PHU_Location_dimension.insert(0, "PHU_surrogate_key", np.arange(len(PHU_Location_dimension)))
PHU_Location_dimension.to_csv("PHU_Location_dimension.csv", index=False)
```

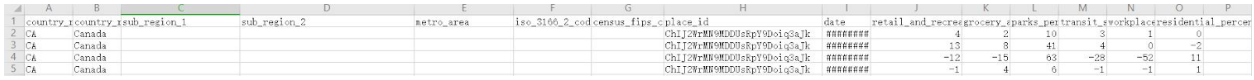
You can see when we meet a new PHU, we will save it in our list, and next time we meet it, we will ignore it.

C8 fx Hamilton Public Health Services

	A	B	C	D	E	F	G	H	I	J
1	PHU_surrogate_key	PHU_id	PHU_name	Address	City	Postal_Cc	Province	URL	Latitude	Longitude
2	0	2241	Kingston, Frontenac and Lennox & Addington Public Heal	221 Portsmouth Avenue	Kingston	K7M 1V5	ON	www.kflap	44.22787	-76.5252
3	1	2253	Peel Public Health	7120 Hurontario Street	Mississauga	L5W 1N4	ON	www.peelp	43.64747	-79.7089
4	2	2236	Halton Region Health Department	1151 Bronte Road	Oakville	L6M 3L1	ON	www.halto	43.414	-79.7448
5	3	2233	Grey Bruce Health Unit	101 17th Street East	Owen Sound	N4K 0A5	ON	www.publi	44.5762	-80.941
6	4	2266	Wellington-Dufferin-Guelph Public Health	160 Chancellors Way	Guelph	N1G 0E1	ON	www.wdgpu	43.52488	-80.2337
7	5	3895	Toronto Public Health	277 Victoria Street, 5th	Toronto	M5B 1W2	ON	www.toror	43.65659	-79.3794
8	6	2237	Hamilton Public Health Services	110 King St. West, 2nd F	Hamilton	L8P 4S6	ON	www.hamil	43.25763	-79.8713
9	7	2246	Niagara Region Public Health Department	1815 Sir Isaac Brock Way	Thorold	L2V 4T7	ON	www.niaga	43.11654	-79.2412
10	8	2270	York Region Public Health Services	17250 Yonge Street	Newmarket	L3Y 6Z1	ON	www.york.	44.04802	-79.4802
11	9	2265	Region of Waterloo, Public Health	99 Regina Street South	Waterloo	N2J 4V3	ON	www.regio	43.46288	-80.5209
12	10	2260	Simcoe Muskoka District Health Unit	15 Sperling Drive	Barrie	L4M 6K9	ON	www.simcc	44.41071	-79.6863
13	11	2256	Porcupine Health Unit	169 Pine Street South	Timmins	P4N 8B7	ON	www.porcu	48.47251	-81.3288
14	12	2230	Durham Region Health Department	605 Rossland Road East	Whitby	L1N 0B2	ON	www.durha	43.89861	-78.9403
15	13	2247	North Bay Parry Sound District Health Unit	345 Oak Street West	North Bay	P1B 2T2	ON	www.myhea	46.31321	-79.4678
16	14	2242	Lambton Public Health	160 Exmouth Street	Point Edward	N7T 7Z6	ON	www.lambt	42.98642	-82.4048
17	15	2257	Renfrew County and District Health Unit	7 International Drive	Pembroke	K8A 6W5	ON	www.rcdhu	45.79941	-77.1187
18	16	2262	Thunder Bay District Health Unit	999 Balmoral Street	Thunder Bay	P7B 6E7	ON	www.tbdhu	48.40057	-89.2589
19	17	2235	Haliburton, Kawartha, Pine Ridge District Health Unit	200 Rose Glen Road	Port Hope	L1A 3V6	ON	www.hkpr.	43.96817	-78.2858
20	18	2251	Ottawa Public Health	100 Constellation Drive	Ottawa	K2G 6J8	ON	www.ottaw	45.34567	-75.7639
21	19	2244	Middlesex-London Health Unit	50 King Street	London	N6A 5L7	ON	www.healt	42.98147	-81.254
22	20	2234	Haldimand-Norfolk Health Unit	12 Gilbertson Drive	Simcoe	N3Y 4N5	ON	www.hnhu.	42.84783	-80.3038
23	21	2268	Windsor-Essex County Health Unit	1005 Ouellette Avenue	Windsor	N9A 4J8	ON	www.wechu	42.3088	-83.0337
24	22	2255	Peterborough Public Health	185 King Street	Peterborough	K9J 2R8	ON	www.peter	44.30163	-78.3213
25	23	2249	Northwestern Health Unit	210 First Street North	Kenora	P9N 2K4	ON	www.nwhu.	49.76961	-94.4883
26	24	4913	Southwestern Public Health	1230 Talbot Street	St. Thomas	N5P 1G9	ON	www.swpub	42.7778	-81.1512
27	25	2238	Hastings and Prince Edward Counties Health Unit	179 North Park Street	Belleville	K8P 4P1	ON	www.hpePu	44.18667	-77.3914
28	26	2243	Leeds, Grenville and Lanark District Health Unit	458 Laurier Boulevard	Brockville	K6V 7A3	ON	www.healt	44.61584	-75.7028
29	27	5183	Huron Perth District Health Unit	653 West Gore Street	Stratford	N5A 1L4	ON	www.hpph.	43.36866	-81.0019
30	28	2258	Eastern Ontario Health Unit	1000 Pitt Street	Cornwall	K6J 5T1	ON	www.eohu.	45.02915	-74.7363
31	29	2227	Brant County Health Unit	194 Terrace Hill Street	Brantford	N3R 1G7	ON	www.bchu.	43.15181	-80.2744
32	30	2240	Chatham-Kent Health Unit	435 Grand Avenue West	Chatham	N7M 5L8	ON	www.ckphu	42.40386	-82.2086
33	31	2226	Algoma Public Health Unit	294 Willow Avenue	Sault Ste. Mari	P6B 0A9	ON	www.algom	46.53237	-84.3148
34	32	2261	Sudbury & District Health Unit	1300 Paris Street	Sudbury	P3E 3A3	ON	www.phsd.	46.46609	-80.9981
35	33	2263	Timiskaming Health Unit	247 Whitewood Avenue, Un	New Liskeard	P0J 1P0	ON	www.timis	47.50928	-79.6816

4: Staging of mobility dimension

This part is a little bit difficult because we should first figure out what values are missing and how to fix them. We first decided



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	country_r	country_n	sub_region_1	sub_region_2	metro_area	iso_3166_2_cod	census_fips_c	place_id	date	retail_and_recre	grocery_pharks_per	transit_s	workplace	residential_percen		
2	CA	Canada						ChIJ2WrMN9MDDUsRpY9Doiq3aJk	########	4	2	10	3	1	0	
3	CA	Canada						ChIJ2WrMN9MDDUsRpY9Doiq3aJk	########	13	8	41	4	0	-2	
4	CA	Canada						ChIJ2WrMN9MDDUsRpY9Doiq3aJk	########	-12	-15	63	-28	-52	11	
5	CA	Canada						ChIJ2WrMN9MDDUsRpY9Doiq3aJk	########	-1	4	6	-1	-1	1	

If the region of the data is missing, we will just delete them, or to say not record them because we need at least the location and date attribute to find the corresponding data when we create a fact table.

	J	K	L	M	N	O	P
7	-4	13			-31	6	
3	-5	19			-30	8	
9	2	21			-30	5	
#	1	25			-24	3	
#	9	33			-5		
#	0	22			-12		
#	3	18			-29	4	

Then we decide to fill the NaN blank to 0. Firstly, there is too much data missing these kinds of values, we cannot delete all of them or the data are very less. And we think the blank should mean there is no record (missing or just no action). If they are missing, we have no idea what value is so we ignore this selection. If it just means no action on that day, the missing value then can be seen as 0 (no action), and that would be easy for us to rewrite.

```python
for idx, row in df.iterrows():
    if (row['sub_region_2'] == "Ottawa Division" or
        row['sub_region_2'] == "Toronto Division" or
        row['sub_region_2'] == "Regional Municipality of Durham" or
        row['sub_region_2'] == "Regional Municipality of Halton" or
        row['sub_region_2'] == "Regional Municipality of Peel" or
        row['sub_region_2'] == "Regional Municipality of York") and (
            int(row["date"][0:4]) == 2020 and int(row["date"][5:7]) >= 9):
        if row['sub_region_2'] == "Ottawa Division":
            row['sub_region_2'] = 'Ottawa'
        if row['sub_region_2'] == "Toronto Division":
            row['sub_region_2'] = 'Toronto'
        if row['sub_region_2'] == "Regional Municipality of Durham":
            row['sub_region_2'] = 'Durham'
        if row['sub_region_2'] == "Regional Municipality of Halton":
            row['sub_region_2'] = 'Halton'
        if row['sub_region_2'] == "Regional Municipality of Peel":
            row['sub_region_2'] = 'Peel'
        if row['sub_region_2'] == "Regional Municipality of York":
            row['sub_region_2'] = 'York'
        mobility_row = [(row["place_id"] + row["date"]).replace("-", ""), row['date'], row['metro_area'],
                        row['sub_region_2'], row['sub_region_1'],
                        row['retail_and_recreation_percent_change_from_baseline'],
                        row['grocery_and_pharmacy_percent_change_from_baseline'],
                        row['parks_percent_change_from_baseline'],
                        row['transit_stations_percent_change_from_baseline'],
                        row['workplaces_percent_change_from_baseline'],
                        row['residential_percent_change_from_baseline']]
```

Next step, since we are focusing on the GTA region and Ottawa area, we need to select only the data with these regions, and we just save the name of this region to meet the type in covid csv for easy searching work later.

```python
Mobility_dimension.insert(0, "Mobility_surrogate_key", np.arange(len(Mobility_dimension)))
Mobility_dimension = Mobility_dimension.fillna({'Retail_and_recreation': '0'})
Mobility_dimension = Mobility_dimension.fillna({'Grocery_and_pharmacy': '0'})
Mobility_dimension = Mobility_dimension.fillna({'Park': '0'})
Mobility_dimension = Mobility_dimension.fillna({'Transit_stations': '0'})
Mobility_dimension = Mobility_dimension.fillna({'Workplaces': '0'})
Mobility_dimension = Mobility_dimension.fillna({'Residential': '0'})
Mobility_dimension.to_csv("Mobility_dimension.csv", index=False)
```

(fill the NaN value to 0)

	Mobility_surrogate_key	Mobility_Mobility_Date	Metro_area	Subregion	Province	Retail_and_recreation	Grocery_and_pharmacy	Park	Transit_stations	Workplaces	Residential
1	0	ChIJmcB2g	2020/9/1		Ottawa	Ontario	-18	-4	47	-59	-57	16
2	1	ChIJmcB2g	2020/9/2		Ottawa	Ontario	-22	-6	8	-62	-57	17
3	2	ChIJmcB2g	2020/9/3		Ottawa	Ontario	-13	2	77	-56	-56	15
4	3	ChIJmcB2g	2020/9/4		Ottawa	Ontario	-22	-5	61	-54	-53	14
5	4	ChIJmcB2g	2020/9/5		Ottawa	Ontario	-18	-5	0	-40	-4	1
6	5	ChIJmcB2g	2020/9/6		Ottawa	Ontario	-2	9	0	-39	4	-2
7	6	ChIJmcB2g	2020/9/7		Ottawa	Ontario	-57	-55	29	-77	-84	21
8	7	ChIJmcB2g	2020/9/8		Ottawa	Ontario	-17	6	13	-58	-58	16
9	8	ChIJmcB2g	2020/9/9		Ottawa	Ontario	-24	-7	2	-61	-57	18

(sample)

5: Staging of weather dimension

In this part, data is not given to us. I firstly download the data files for the 6 regions we focus on, which is quite simple.
For the missing values handling, I firstly decided to fill all of them with 0 because they are all numeric data. However, some of them belong to max temperature and min temperature. If I just wrote 0 there, the analysis will be interfered. Therefore, I choose to ignore them. As for the amount of rain, snow and total precipitation, I regard it as no precipitation and fill in with 0.

```
#%%

area_climate = area_climate.fillna({'Total Rain (mm)': '0.0'})  area_climate: {DataFrame: (732, 13)}
area_climate = area_climate.fillna({'Total Snow (cm)': '0.0'})  area_climate: {DataFrame: (732, 13)}
area_climate = area_climate.fillna({'Total Precip (mm)': '0.0'})  area_climate: {DataFrame: (732, 13)}
```

```
#%%

area_climate['Weather Key'] = area_climate['Weather Key'].str.replace('-', '')  area_climate: {DataFrame: (732, 13)}  ar
area_climate.replace("OTTAWA CDA", "Ottawa", inplace = True)  area_climate: {DataFrame: (732, 13)}
area_climate.replace("TORONTO CITY CENTRE", "Toronto", inplace = True)  area_climate: {DataFrame: (732, 13)}
area_climate.replace("TORONTO NORTH YORK", "Newmarket", inplace = True)  area_climate: {DataFrame: (732, 13)}
area_climate.replace("OSHAWA", "Whitby", inplace = True)  area_climate: {DataFrame: (732, 13)}
area_climate.replace("TORONTO INTL A", "Mississauga", inplace = True)  area_climate: {DataFrame: (732, 13)}
area_climate.replace("OAKVILLE TWN", "Oakville", inplace = True)  area_climate: {DataFrame: (732, 13)}
```

Because we are asked to map PHU, Mobility and weather by location, I have to make sure they can be matched. To do that, I manually found the association between the weather states, the regions and the cities, and changed the location name to the cities that appeared in the Confirmed_Positive_Cases_ON.csv file.

```
#%%

area_climate.columns = ['surrogate_key', 'Longitude', 'Latitude', 'Station_Name', 'Climate_ID', 'Date_Time',  area_clima
       'Max_Temp', 'Min_Temp',  'Mean_Temp',
       'Total_Rain_mm',
       'Total_Snow_cm',  'Total_Precip_mm', 'Weather_Key']
```

Lastly, I change the headers to the string that can be accepted by psql, which do not contain special characters and spaces.

6: Map PHU, Mobility, and weather dimension

PHU:

	A	B	C	D	E	F	G	H	I	J	K
1	PHU_surrogate_key	PHU_id	PHU_name	Address	City	Postal_Cc	Province	URL	Latitude	Longitude	
2	0	2241	Kingston, Frontenac and Lennox & Addington Public Heal	221 Portsmouth Avenue	Kingston	K7M 1V5	ON	www.kflap	44.22787	-76.5252	
3	1	2253	Peel Public Health	7120 Hurontario Street	Mississauga	L5W 1N4	ON	www.peelr	43.64747	-79.7089	
4	2	2236	Halton Region Health Department	1151 Bronte Road	Oakville	L6M 3L1	ON	www.halto	43.414	-79.7448	
5	3	2233	Grey Bruce Health Unit	101 17th Street East	Owen Sound	N4K 0A5	ON	www.publi	44.5762	-80.941	

Mobility:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Mobility_	Mobility_key	Date	Metro-are	Subregior	Province	Retail_and_recreation	Grocery_and_pharmacy	Park	Transit_stations	Workplaces	Residential
2	0	ChIJmcB2guEn0UwRrORV_iicOL	2020/9/1		Ottawa	Ontario	-18	-4	47	-59	-57	16
3	1	ChIJmcB2guEn0UwRrORV_iicOL	2020/9/2		Ottawa	Ontario	-22	-6	8	-62	-57	17
4	2	ChIJmcB2guEn0UwRrORV_iicOL	2020/9/3		Ottawa	Ontario	-13	2	77	-56	-56	15
5	3	ChIJmcB2guEn0UwRrORV_iicOL	2020/9/4		Ottawa	Ontario	-22	-5	61	-54	-53	14

Weather:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	surrogate	Longitude	Latitude	Station Name	Climate I	Date/Time	Max Temp	Min Temp	Mean Temp	Total Rai	Total Snc	Total Pre	Weather Key
2	0	-75.72	45.38	Ottawa	6105976	2020-09-01	25	13	19	0	0	0	610597620200901
3	1	-75.72	45.38	Ottawa	6105976	2020-09-02	27	19.5	23.3	7	0	7	610597620200902
4	2	-75.72	45.38	Ottawa	6105976	2020-09-03	24.5	14	19.3	2	0	2	610597620200903
5	3	-75.72	45.38	Ottawa	6105976	2020-09-04	21	12	16.5	0.4	0	0.4	610597620200904

To map these three dimensions, we should know how to present them as one. On a day, in a location, a PHU in that location, the weather is xxx, the daily mobility is xxx. The daily information for mobility and weather is different, but PHU is never changed. So first we make sure both mobility and weather have its date attribute, and all of them three have location attributes. We see the mobility just has province and region, the PHU has region and city, and weather has city. So we can first link PHU and weather by city, we will get in a “period of time, in a region”, and link it to mobility with the date and region since mobility has date and location attribute. So we now with the current attribute, we can make sure on a day, in a location, the PHU, weather and mobility dimensions are linked.

7: Staging of special measures dimension

Special measure is a tricky part. The biggest issue I faced is to find the measures. I firstly misunderstood how I am asked to structure the information. I just found 10 measures in different fields including policies, vaccines, etc. However, when we moved onto the fact table, we realized that it cannot be matched with other data because their timelines overlapped and did not have location variables. Then I sent emails to the TA and knew that I should find more general measures such as State 2, State 3 and Colour coded system to separate the

timeline. I also divided them by cities and extended the description to several columns to make it more detailed. The below is the final source file of measures.

Title	City	Start date	End date	Private indoc	Private outdc	Public indoor	Public outdo	Indoor religic	Outdoor relig	mask requirc	Self-isolatin	Essential ser	Entertainme	School/work	Restaurant patrons seated indoors
Stage 3	Ottawa	2020-07-17	2020-10-09	50	100	50	100	50	100	Yes	14 days	Open	Has constrai	Online	N/A
Stage 3	York	2020-07-24	2020-10-18	50	100	50	100	50	100	Yes	14 days	Open	Has constrai	Online	N/A
Stage 3	Peel	2020-07-31	2020-10-09	50	100	50	100	50	100	Yes	14 days	Open	Has constrai	Online	N/A
Stage 3	Toronto	2020-07-31	2020-10-09	50	100	50	100	50	100	Yes	14 days	Open	Has constrai	Online	N/A
Stage 3	Durham	2020-07-24	2020-11-06	50	100	50	100	50	100	Yes	14 days	Open	Has constrai	Online	N/A
Stage 3	Halton	2020-07-24	2020-11-06	50	100	50	100	50	100	Yes	14 days	Open	Has constrai	Online	N/A
Modified Sta	Ottawa	2020-10-10	2020-11-06	10	25	10	25	10	25	Yes	14 days	Open	Has constrai	Online	N/A
Modified Sta	York	2020-10-19	2020-11-06	10	25	10	25	10	25	Yes	14 days	Open	Has constrai	Online	N/A
Modified Sta	Peel	2020-10-10	2020-11-06	10	25	10	25	10	25	Yes	14 days	Open	Has constrai	Online	N/A
Modified Sta	Toronto	2020-10-10	2020-11-06	10	25	10	25	10	25	Yes	14 days	Open	Has constrai	Online	N/A
Yellow Zone	Durham	2020-11-07	2020-11-15	10	25	50	100	30 percent c	100	Yes	14 days	Open	Has constrai	Online	N/A
Yellow Zone	Halton	2020-11-07	2020-11-15	10	25	50	100	30 percent c	100	Yes	14 days	Open	Has constrai	Online	N/A
Orange Zone	Ottawa	2020-11-07	2020-12-25	10	25	50	100	30 percent c	100	Yes	14 days	Open	Has constrai	Online	50
Orange Zone	York	2020-11-07	2020-11-15	10	25	50	100	30 percent c	100	Yes	14 days	Open	Has constrai	Online	50
Red Zone	Peel	2020-11-07	2020-12-20	0	0	5	25	30 percent c	100	Yes	14 days	Open	Has constrai	Online	10
Orange Zone	Toronto	2020-11-07	2020-11-13	10	25	50	100	30 percent c	100	Yes	14 days	Open	Has constrai	Online	50
Orange Zone	Durham	2020-11-16	2020-12-20	10	25	50	100	30 percent c	100	Yes	14 days	Open	Has constrai	Online	50
Red Zone	Halton	2020-11-16	2020-12-25	0	0	5	25	30 percent c	100	Yes	14 days	Open	Has constrai	Online	10
Grey Zone	Ottawa	2020-12-26	2021-01-23	0	0	0	10	10	10	Yes	14 days	Open	Closure	Online	Closure
Red Zone	York	2020-11-16	2020-12-13	0	0	5	25	30 percent c	100	Yes	14 days	Open	Has constrai	Online	10
Grey Zone	Peel	2020-12-21	2021-01-23	0	0	0	10	10	10	Yes	14 days	Open	Closure	Online	Closure
Red Zone	Toronto	2020-11-14	2020-12-20	0	0	5	25	30 percent c	100	Yes	14 days	Open	Has constrai	Online	10
Red Zone	Durham	2020-12-21	2020-12-25	0	0	5	25	30 percent c	100	Yes	14 days	Open	Has constrai	Online	10
Grey Zone	Halton	2020-12-26	2021-01-23	0	0	0	10	10	10	Yes	14 days	Open	Closure	Online	Closure
Grey Zone	York	2020-12-14	2021-01-23	0	0	0	10	10	10	Yes	14 days	Open	Closure	Online	Closure
Grey Zone	Toronto	2020-12-21	2021-01-23	0	0	0	10	10	10	Yes	14 days	Open	Closure	Online	Closure
Grey Zone	Durham	2020-12-26	2021-01-23	0	0	0	10	10	10	Yes	14 days	Open	Closure	Online	Closure

In terms of dimension creation, it is kind of the same as the weather dimension creation, which is another part I did. I added the surrogate key and changed the header.

8: Surrogate key pipeline – including role-playing dates

The surrogate keys of each dimension have been shown in the previous picture.

```
Specimen_date_dimension = pd.DataFrame(
    columns=['Specimen_date_key', 'Date', 'Day', 'Month', 'Day_of_Week',
             'Week_in_Year', 'Weekend', 'Holiday', 'Season'])

for idx, row in date.iterrows():
    date_row = ["Sp"+row["id"].replace("-", ""), row["id"], row['Day'], row['Month'], row['Day_of_Week'],
                row['Week_in_Year'], row['Weekend'], row['Holiday'], row['Season']]
    Specimen_date_dimension.loc[len(Specimen_date_dimension)] = date_row

Specimen_date_dimension.insert(0, "Specimen_date_surrogate_key", np.arange(len(Specimen_date_dimension)))
Specimen_date_dimension.to_csv("Specimen_Date_dimension.csv", index=False)
```

We make sure we first deal with the missing values and after the dimension is created, we add the surrogate key to make the final dimension.

9: Staging of fact table – including FKs and measures

This part is the most difficult one and we find lots of problems that we did not find when we were staging other dimensions. First I noticed that for a dimension like PHU, which has a clear “ID” to make a match, it is easy to code. We just need to compare the PHU ID Covid csv and PHU ID in PHU dimension then we can add PHU KEY in our fact table. But for some dimensions, there is no obvious value that can be directly matched such as special measures, patient. So I spend lots of time selecting the most suitable attribute to match the key. Like use reported date and PHU location to match the mobility dimension, use row_id as the patient id (since row-id and patient id are both unique).

When I start coding, I find that the code in the given python problem is not working.
“Dimension[Dimension[‘x’]==row[‘x]][‘key’].values[0]”.
It shows an overflow error, I try to fix the problem and do some research on it but cannot get a solution. All i know is when I type values, it will show the value and its dtype, and when I add [0], the error occurs. Therefore, I use another way to get the exact value.

```
Onset_Date_dimension[Onset_Date_dimension["Onset_date_key"] ==
                    row["Onset_date_key"]]["Onset_date_surrogate_key"].to_string(index=False),
```

Change .values[0] to to_string(index=False).
I know the [0] should mean removing the index part of the value, so the index=False will do the same function.

After that, we set our range to find the required data, between 2020-09-01 to 2020-12-31 in the GTA region and Ottawa area.

```
if int(row['Accurate_Episode_Date'][0:4]) == 2020 \
        and int(row['Accurate_Episode_Date'][5:7]) >= 9 \
        and int(row['Case_Reported_Date'][0:4]) == 2020 \
        and str(row['Test_Reported_Date']) != "nan" \
        and str(row['Specimen_Date']) != "nan" \
        and (row['Reporting_PHU'] == "Ottawa Public Health" or
             row['Reporting_PHU'] == "Toronto Public Health" or
             row['Reporting_PHU'] == "Durham Region Health Department" or
             row['Reporting_PHU'] == "Halton Region Health Department" or
             row['Reporting_PHU'] == "Peel Public Health" or
             row['Reporting_PHU'] == "York Region Public Health Services"):
```

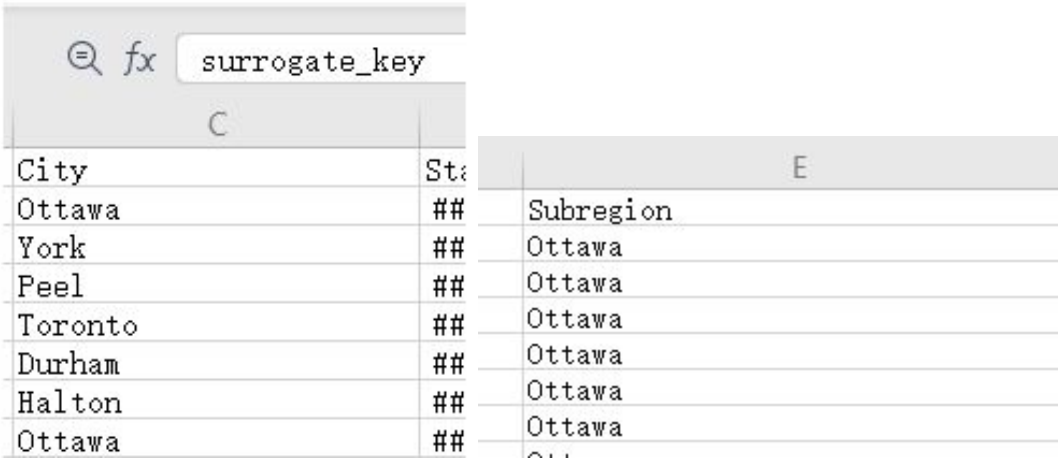
We know the accurate date is always earlier than the other date, so we just need to set the accurate date in 2020 and later than Sep. By the way, since we are using case reported dates for matching weather, mobility and special measures, we need also make sure the case reported date is in 2020. (like if the accurate date is 2020/12/31, and the reported date is 2021/1/1, that is not what we need). And I found the test reported date and specimen date in some row are missing. But there are very few of the data missing these two values and if the dates are missing, we cannot add the dates by yourself since we don't know if they have a report or not, therefore I decide just delete the data if they are missing these two values. Next, we set

the location in the GTA region and ottawa area,

```
if row['Reporting_PHU'] == "Ottawa Public Health":
    row['Reporting_PHU'] = "Ottawa"
if row['Reporting_PHU'] == "Toronto Public Health":
    row['Reporting_PHU'] = "Toronto"
if row['Reporting_PHU'] == "Durham Region Health Department":
    row['Reporting_PHU'] = "Durham"
if row['Reporting_PHU'] == "Halton Region Health Department":
    row['Reporting_PHU'] = "Halton"
if row['Reporting_PHU'] == "Peel Public Health":
    row['Reporting_PHU'] = "Halton"
if row['Reporting_PHU'] == "York Region Public Health Services":
    row['Reporting_PHU'] = "York"
```

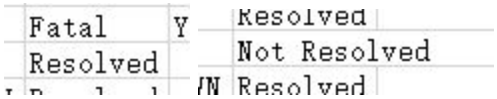
If the data match the region, we change the region to a convenient type “Region” to fit the type in other dimensions. (like to match mobility key and special measure key)

```
fact_row = [Onset_Date_dimension[Onset_Date_dimension['Date'] ==
                                 row['Accurate_Episode_Date'][:10]]['Onset_date_key'].to_string(index=False),
            Reported_Date_dimension[Reported_Date_dimension['Date'] ==
                                    row['Case_Reported_Date'][:10]]['Reported_date_key'].to_string(index=False),
            Test_Date_dimension[Test_Date_dimension['Date'] ==
                                row['Test_Reported_Date'][:10]]['Test_date_key'].to_string(index=False),
            Specimen_Date_dimension[Specimen_Date_dimension['Date'] ==
                                    row['Specimen_Date'][:10]]['Specimen_date_key'].to_string(index=False),
            row['Row_ID'],
            row['Reporting_PHU_ID'],
            Mobility_dimension[(Mobility_dimension['Subregion'] == row['Reporting_PHU']) &
                               (Mobility_dimension['Date'] == row['Case_Reported_Date'][:10])]
            ['Mobility_key'].to_string(index=False),
            weather_dimension[(weather_dimension["Date/Time"] == row['Case_Reported_Date'][:10]) &
                              (weather_dimension["Station Name"] == row['Reporting_PHU_City'])]
            ["Weather Key"].to_string(index=False),
            Special_Measures_dimension[(Special_Measures_dimension["City"] == row['Reporting_PHU']) &
                                       (Special_Measures_dimension['Start date'] < row['Case_Reported_Date'][:10]) &
                                       (Special_Measures_dimension['End date'] > row['Case_Reported_Date'][:10])]
            ["Special Measures Key"].to_string(index=False)
            ]
```



Then we add measures in fact table

```
if row['Outcome1'] == "Resolved":
    fact_row += ["Yes", "No", "No"]
if row['Outcome1'] == "Not Resolved":
    fact_row += ["No", "Yes", "No"]
if row['Outcome1'] == "Fatal":
    fact_row += ["No", "No", "Yes"]
```



In the given csv, we can only know if the patient is resolved, not resolved or fatal.

Finally, we get all the keys from the dimension, we now need to change to key to surrogate key.

```
for idx, row in fact_table.iterrows():
    fact_row = [
        Onset_Date_dimension[Onset_Date_dimension["Onset_date_key"] ==
                             row["Onset_date_key"]]["Onset_date_surrogate_key"].to_string(index=False),
        Reported_Date_dimension[Reported_Date_dimension["Reported_date_key"] ==
                                row["Reported_date_key"]]["Reported_date_surrogate_key"].to_string(index=False),
        Test_Date_dimension[Test_Date_dimension["Test_date_key"] ==
                            row["Test_date_key"]]["Test_date_surrogate_key"].to_string(index=False),
        Specimen_Date_dimension[Specimen_Date_dimension["Specimen_date_key"] ==
                                row["Specimen_date_key"]]["Specimen_date_surrogate_key"].to_string(index=False),
        patient_dimension[patient_dimension["Patient_Key"] ==
                          row["Patient_Key"]]["Patient_surrogate_key"].to_string(index=False),
        PHU_Location_dimension[PHU_Location_dimension["PHU_id"] ==
                               row["PHU_id"]]["PHU_surrogate_key"].to_string(index=False),
        Mobility_dimension[Mobility_dimension["Mobility_key"] ==
                           row["Mobility_key"]]["Mobility_surrogate_key"].to_string(index=False),
        weather_dimension[weather_dimension["Weather Key"] ==
                          row["Weather Key"]]["surrogate_key"].to_string(index=False),
        Special_Measures_dimension[Special_Measures_dimension["Special Measures Key"] ==
                                   row["Special Measures Key"]]["surrogate_key"].to_string(index=False),
        row["Resolved"],
        row["Unresolved"],
        row["Fatal"]
```

The code is almost the same as the example python(just change the values[0]).

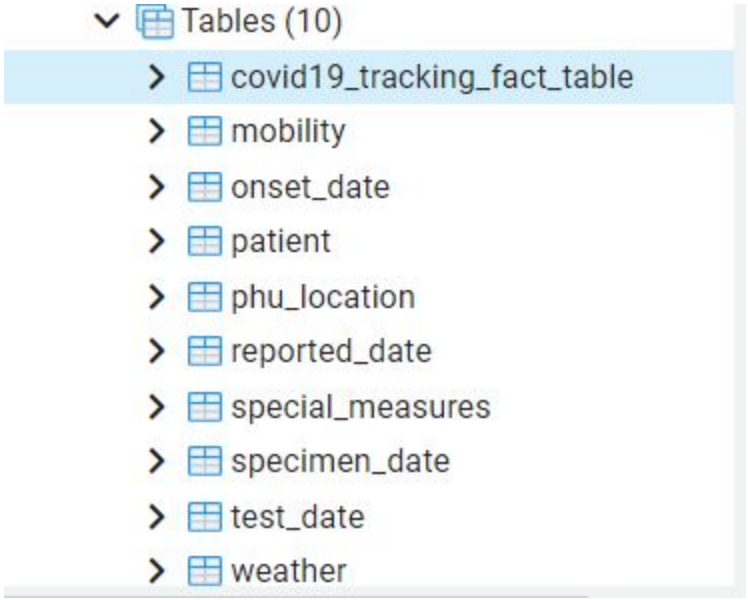
```
    final_fact_table.loc[len(final_fact_table)] = fact_row

final_fact_table = final_fact_table.replace('Series([], )', 'Nomatch')
final_fact_table.to_csv("final_fact_table.csv", index=False)
```

Then we output the final_fact_table with surrogate keys, and we finish all the steps for the fact table :)

10: Create database instance and tables

In this part, we first connect to the group database in pgAdmin. Then, import the sql file in the query tool and run to create tables in the database. Then, we use “import/export” to import the processed data from csv to each table.



11: DataStaging team planning

Below is how the tasks divided in the team. For a clearer view, please see the “DataStagingTeamPlanning.xlsx”.

CSI4142 Physical Design						
Deliverable	**Team member(s) responsible**	**Expected completion date**	**Actual completion date**	**Estimate time (hours) to complete**	**Actual time (hours) to complete**	**Notes (if any)**
Create database instance and tables	Yiwen Liu	3-9	3-9	2	2	
Staging of Date dimension	Chuhao Jia	2-24	2-24	1	1	Create an individual date dimension and colonel it for others role-playing dates
Staging of Patient dimension	Chuhao Jia	2-24	2-24	1	0.5	Not sure how to use Age group in future so I did not know if I should change the age gourp with symbol "<" and "s" in the end.
Staging of PHU dimension	Chuhao Jia	2-24	2-24	1	0.1	
Staging of Mobility dimension	Yiwen Liu/Chuhao Jia	2-24	2-24	1	0.1	
Staging of Weather dimension	Xiaohan Yu	2-24	2-24	2	1	
Map PHU, Mobility and Weather dimensions	Chuhao Jia	3-3	3-8	1	1	
Staging of Special Measures dimension	Xiaohan Yu	2-24	3-3	3	5.5	Reseaching takes longer time than planned. I firstly misunderstood how I should structure the measures. After communicating with TA, I redid it. So It took a long time.
Surrogate key pipeline – including role-playing dates	Chuhao Jia	3-3	3-8	1	0.1	
Staging of fact table – including FKs and measures	Chuhao Jia	3-3	3-8	1	4	The function of Values[0] occur overflow error but I cannot figure out the reason so I change another way to get values
Data quality handling and reporting	Chuhao Jia	3-8	3-9	1	1	Not all the missing value are fixed. Because I did not know what should we do with all of these value in the next stage. So I just fix the value that I think I understand how would we use in the future
Others - if any						