Group 4

Final Project

by

Michael Zeolla

Samrat Korupolu,

Tatyanna Villegas,

You Zhou

Contents

Executive Summary

Our team's objective is to build a Covid-19 Case Database which would include the following details: County, Cases, Case Count, Case Location, Hospital Beds, Mask Use, etc.

Various sources (such as NY Times, John Hopkins, FL Department of Health) have collected and made data publicly available regarding the pandemic. We will be focusing on local data regarding Florida counties on the number of current cases, test results, reported deaths, and recoveries. Combined with additional information on mask use as well as hospital beds available in each area, this can lead to some interesting queries. Potential areas of exploration include identifying which areas have been most affected, which areas are hotspots with current cases, or which areas are currently at-risk considering bed capacity and mask use.

Group members: You Zhou, Tatyanna Villegas, Samrat Korupolu, Michael Zeolla

Part 1: Database Design

Case Description

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

Section 1.1 Data Integrity

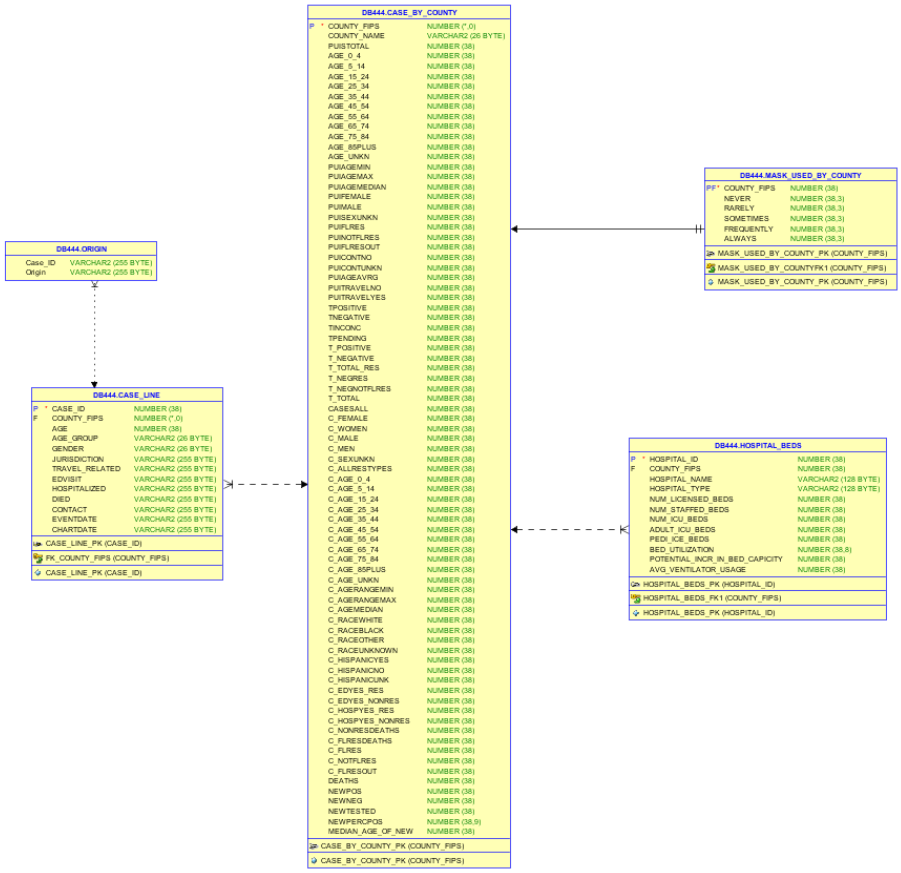
1.1.1. Data set choice and basic dimensions

There are four main datasets we are using in the project. Three of them are derived from the Florida Department of Health Open Data. (Data time we used is up to Oct.5th)

https://open-fdoh.hub.arcgis.com/datasets/florida-covid19-case-line-data

Dataset	Rows	Columns	Description
Case by County	68	81	Record cases by county level totals
Mask use by county	68	6	Mask used by county level
Case Line	726014	13	Basic case related information in FL
Hospital Beds	275	12	The Hospital Bed status data
Origins	737023	2	Origin dataset with related Case_ID Number

1.1.2 ERD



1.1.3 Data Dictionary

CASE_BY_COUNTY

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
COUNTY_FIPS	NUMBER(38,0)	No	null	1	null
COUNTY_NAME	VARCHAR2(26 BYTE)	Yes	null	2	null
PUISTOTAL	NUMBER(38,0)	Yes	null	3	null

AGE_0_4	NUMBER(38,0)	Yes	null	4	null
AGE_5_14	NUMBER(38,0)	Yes	null	5	null
AGE_15_24	NUMBER(38,0)	Yes	null	6	null
AGE_25_34	NUMBER(38,0)	Yes	null	7	null
AGE_35_44	NUMBER(38,0)	Yes	null	8	null
AGE_45_54	NUMBER(38,0)	Yes	null	9	null
AGE_55_64	NUMBER(38,0)	Yes	null	10	null
AGE_65_74	NUMBER(38,0)	Yes	null	11	null
AGE_75_84	NUMBER(38,0)	Yes	null	12	null
AGE_85PLUS	NUMBER(38,0)	Yes	null	13	null
AGE_UNKN	NUMBER(38,0)	Yes	null	14	null
PUIAGEMIN	NUMBER(38,0)	Yes	null	15	null
PUIAGEMAX	NUMBER(38,0)	Yes	null	16	null
PUIAGEMEDIAN	NUMBER(38,0)	Yes	null	17	null
PUIFEMALE	NUMBER(38,0)	Yes	null	18	null
PUIMALE	NUMBER(38,0)	Yes	null	19	null
PUISEXUNKN	NUMBER(38,0)	Yes	null	20	null
PUIFLRES	NUMBER(38,0)	Yes	null	21	null
PUINOTFLRES	NUMBER(38,0)	Yes	null	22	null
PUIFLRESOUT	NUMBER(38,0)	Yes	null	23	null
PUICONTNO	NUMBER(38,0)	Yes	null	24	null
PUICONTUNKN	NUMBER(38,0)	Yes	null	25	null
PUIAGEAVRG	NUMBER(38,0)	Yes	null	26	null
PUITRAVELNO	NUMBER(38,0)	Yes	null	27	null

PUITRAVELYES	NUMBER(38,0)	Yes	null	28	null
TPOSITIVE	NUMBER(38,0)	Yes	null	29	null
TNEGATIVE	NUMBER(38,0)	Yes	null	30	null
TINCONC	NUMBER(38,0)	Yes	null	31	null
TPENDING	NUMBER(38,0)	Yes	null	32	null
T_POSITIVE	NUMBER(38,0)	Yes	null	33	null
T_NEGATIVE	NUMBER(38,0)	Yes	null	34	null
T_TOTAL_RES	NUMBER(38,0)	Yes	null	35	null
T_NEGRES	NUMBER(38,0)	Yes	null	36	null
T_NEGNOTFLRES	NUMBER(38,0)	Yes	null	37	null
T_TOTAL	NUMBER(38,0)	Yes	null	38	null
CASESALL	NUMBER(38,0)	Yes	null	39	null
C_FEMALE	NUMBER(38,0)	Yes	null	40	null
C_WOMEN	NUMBER(38,0)	Yes	null	41	null
C_MALE	NUMBER(38,0)	Yes	null	42	null
C_MEN	NUMBER(38,0)	Yes	null	43	null
C_SEXUNKN	NUMBER(38,0)	Yes	null	44	null
C_ALLRESTYPES	NUMBER(38,0)	Yes	null	45	null
C_AGE_0_4	NUMBER(38,0)	Yes	null	46	null
C_AGE_5_14	NUMBER(38,0)	Yes	null	47	null
C_AGE_15_24	NUMBER(38,0)	Yes	null	48	null
C_AGE_25_34	NUMBER(38,0)	Yes	null	49	null
C_AGE_35_44	NUMBER(38,0)	Yes	null	50	null
C_AGE_45_54	NUMBER(38,0)	Yes	null	51	null

C_AGE_55_64	NUMBER(38,0)	Yes	null	52	null
C_AGE_65_74	NUMBER(38,0)	Yes	null	53	null
C_AGE_75_84	NUMBER(38,0)	Yes	null	54	null
C_AGE_85PLUS	NUMBER(38,0)	Yes	null	55	null
C_AGE_UNKN	NUMBER(38,0)	Yes	null	56	null
C_AGERANGEMIN	NUMBER(38,0)	Yes	null	57	null
C_AGERANGEMAX	NUMBER(38,0)	Yes	null	58	null
C_AGEMEDIAN	NUMBER(38,0)	Yes	null	59	null
C_RACEWHITE	NUMBER(38,0)	Yes	null	60	null
C_RACEBLACK	NUMBER(38,0)	Yes	null	61	null
C_RACEOTHER	NUMBER(38,0)	Yes	null	62	null
C_RACEUNKNOWN	NUMBER(38,0)	Yes	null	63	null
C_HISPANICYES	NUMBER(38,0)	Yes	null	64	null
C_HISPANICNO	NUMBER(38,0)	Yes	null	65	null
C_HISPANICUNK	NUMBER(38,0)	Yes	null	66	null
C_EDYES_RES	NUMBER(38,0)	Yes	null	67	null
C_EDYES_NONRES	NUMBER(38,0)	Yes	null	68	null
C_HOSPYES_RES	NUMBER(38,0)	Yes	null	69	null
C_HOSPYES_NONRES	NUMBER(38,0)	Yes	null	70	null
C_NONRESDEATHS	NUMBER(38,0)	Yes	null	71	null
C_FLRESDEATHS	NUMBER(38,0)	Yes	null	72	null

C_FLRES	NUMBER(38,0)	Yes	null	73	null
C_NOTFLRES	NUMBER(38,0)	Yes	null	74	null
C_FLRESOUT	NUMBER(38,0)	Yes	null	75	null
DEATHS	NUMBER(38,0)	Yes	null	76	null
NEWPOS	NUMBER(38,0)	Yes	null	77	null
NEWNEG	NUMBER(38,0)	Yes	null	78	null
NEWTESTED	NUMBER(38,0)	Yes	null	79	null
NEWPERCPOS	NUMBER(38,9)	Yes	null	80	null
MEDIAN_AGE_OF_NEW	NUMBER(38,0)	Yes	null	81	null

CASE_LINE

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
CASE_ID	NUMBER(38,0)	No	null	1	null
COUNTY_FIPS	NUMBER(38,0)	Yes	null	2	null
AGE	NUMBER(38,0)	Yes	null	3	null
AGE_GROUP	VARCHAR2(26 BYTE)	Yes	null	4	null
GENDER	VARCHAR2(26 BYTE)	Yes	null	5	null
JURISDICTION	VARCHAR2(255 BYTE)	Yes	null	6	null
TRAVEL_RELATED	VARCHAR2(255 BYTE)	Yes	null	7	null
EDVISIT	VARCHAR2(255 BYTE)	Yes	null	8	null

HOSPITALIZED	VARCHAR2(255 BYTE)	Yes	null	9	null
DIED	VARCHAR2(255 BYTE)	Yes	null	10	null
CONTACT	VARCHAR2(255 BYTE)	Yes	null	11	null
EVENTDATE	VARCHAR2(255 BYTE)	Yes	null	12	null
CHARTDATE	VARCHAR2(255 BYTE)	Yes	null	13	null

HOSPITAL_BEDS

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
HOSPITAL_ID	NUMBER(38,0)	No	null	1	null
COUNTY_FIPS	NUMBER(38,0)	Yes	null	2	null
HOSPITAL_NAME	VARCHAR2(128 BYTE)	Yes	null	3	null
HOSPITAL_TYPE	VARCHAR2(128 BYTE)	Yes	null	4	null
NUM_LICENSED_BEDS	NUMBER(38,0)	Yes	null	5	null
NUM_STAFFED_BEDS	NUMBER(38,0)	Yes	null	6	null
NUM_ICU_BEDS	NUMBER(38,0)	Yes	null	7	null
ADULT_ICU_BEDS	NUMBER(38,0)	Yes	null	8	null

PEDI_ICE_BEDS	NUMBER(38,0)	Yes	null	9	null
BED_UTILIZATION	NUMBER(38,8)	Yes	null	10	null
POTENTIAL_INCR_IN_BED_CAPICITY	NUMBER(38,0)	Yes	null	11	null
AVG_VENTILATOR_USAGE	NUMBER(38,0)	Yes	null	12	null

MASK_USED_BY_COUNTY

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
COUNTY_FIPS	NUMBER(38,0)	No	null	1	null
NEVER	NUMBER(38,3)	Yes	null	2	null
RARELY	NUMBER(38,3)	Yes	null	3	null
SOMETIMES	NUMBER(38,3)	Yes	null	4	null
FREQUENTLY	NUMBER(38,3)	Yes	null	5	null
ALWAYS	NUMBER(38,3)	Yes	null	6	null

ORIGIN

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
Case_ID	VARCHAR2(255 BYTE)	No	null	1	null
Origin	VARCHAR2(255 BYTE)	Yes	null	2	null

Section 1.2: Data Generation and Loading

1.2.1 Table Creation

Create Table : Case_By_County

Result:

```
CREATE TABLE "DB444"."CASE_BY_COUNTY"
  (        "COUNTY_FIPS" NUMBER(*,0),
           "COUNTY_NAME" VARCHAR2(26 BYTE),
           "PUISTOTAL" NUMBER(38,0),
           "AGE_0_4" NUMBER(38,0),
           "AGE_5_14" NUMBER(38,0),
           "AGE_15_24" NUMBER(38,0),
           "AGE_25_34" NUMBER(38,0),
           "T_NEGATIVE" NUMBER(38,0),
           "T_TOTAL_RES" NUMBER(38,0),
           … … … … … … …
           "C_NOTFLRES" NUMBER(38,0),
           "C_FLRESOUT" NUMBER(38,0),
           "DEATHS" NUMBER(38,0),
           "NEWPOS" NUMBER(38,0),
           "NEWNEG" NUMBER(38,0),
           "NEWTESTED" NUMBER(38,0),
           "NEWPERCPOS" NUMBER(38,9),
           "MEDIAN_AGE_OF_NEW" NUMBER(38,0)
  ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COUNTY_FIPS	NUMBER(38,0)	No	(null)	1	(null)
2	COUNTY_NAME	VARCHAR2(26 BYTE)	Yes	(null)	2	(null)
3	PUISTOTAL	NUMBER(38,0)	Yes	(null)	3	(null)
4	AGE_0_4	NUMBER(38,0)	Yes	(null)	4	(null)
5	AGE_5_14	NUMBER(38,0)	Yes	(null)	5	(null)
6	AGE_15_24	NUMBER(38,0)	Yes	(null)	6	(null)
7	AGE_25_34	NUMBER(38,0)	Yes	(null)	7	(null)
8	AGE_35_44	NUMBER(38,0)	Yes	(null)	8	(null)
9	AGE_45_54	NUMBER(38,0)	Yes	(null)	9	(null)
10	AGE_55_64	NUMBER(38,0)	Yes	(null)	10	(null)
11	AGE_65_74	NUMBER(38,0)	Yes	(null)	11	(null)
12	AGE_75_84	NUMBER(38,0)	Yes	(null)	12	(null)
13	AGE_85PLUS	NUMBER(38,0)	Yes	(null)	13	(null)
14	AGE_UNKN	NUMBER(38,0)	Yes	(null)	14	(null)
15	PUIAGEMIN	NUMBER(38,0)	Yes	(null)	15	(null)
16	PUIAGEMAX	NUMBER(38,0)	Yes	(null)	16	(null)
17	PUIAGEMEDIAN	NUMBER(38,0)	Yes	(null)	17	(null)
18	PUIFEMALE	NUMBER(38,0)	Yes	(null)	18	(null)
19	PUIMALE	NUMBER(38,0)	Yes	(null)	19	(null)
20	PUISEXUNKN	NUMBER(38,0)	Yes	(null)	20	(null)
21	PUIFLRES	NUMBER(38,0)	Yes	(null)	21	(null)
22	PUINOTFLRES	NUMBER(38,0)	Yes	(null)	22	(null)
23	PUIFLRESOUT	NUMBER(38,0)	Yes	(null)	23	(null)
24	PUICONTNO	NUMBER(38,0)	Yes	(null)	24	(null)
25	PUICONTUNKN	NUMBER(38,0)	Yes	(null)	25	(null)

Create table: Case_Lin

Result:

```
CREATE TABLE "DB444"."CASE_LINE"
(       "CASE_ID" NUMBER(38,0) NOT NULL ENABLE,
        "COUNTY_FIPS" NUMBER(*,0),
        "AGE" NUMBER(38,0),
        "AGE_GROUP" VARCHAR2(26 BYTE),
        "GENDER" VARCHAR2(26 BYTE),
        "JURISDICTION" VARCHAR2(255 BYTE),
….. ….. ….. ….. ….. …..
        "CONTACT" VARCHAR2(255 BYTE),
        "EVENTDATE" VARCHAR2(255 BYTE),
        "CHARTDATE" VARCHAR2(255 BYTE),
         CONSTRAINT "CASE_LINE_PK" PRIMARY KEY ("CASE_ID")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "STUDENTS"  ENABLE,
         CONSTRAINT "FK_COUNTY_FIPS" FOREIGN KEY ("COUNTY_FIPS")
          REFERENCES "DB444"."CASE_BY_COUNTY" ("COUNTY_FIPS") ENABLE
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CASE_ID	NUMBER(38,0)	No	(null)	1	(null)
2	COUNTY_FIPS	NUMBER(38,0)	Yes	(null)	2	(null)
3	AGE	NUMBER(38,0)	Yes	(null)	3	(null)
4	AGE_GROUP	VARCHAR2(26 BYTE)	Yes	(null)	4	(null)
5	GENDER	VARCHAR2(26 BYTE)	Yes	(null)	5	(null)
6	JURISDICTION	VARCHAR2(255 BYTE)	Yes	(null)	6	(null)
7	TRAVEL_RELATED	VARCHAR2(255 BYTE)	Yes	(null)	7	(null)
8	EDVISIT	VARCHAR2(255 BYTE)	Yes	(null)	8	(null)
9	HOSPITALIZED	VARCHAR2(255 BYTE)	Yes	(null)	9	(null)
10	DIED	VARCHAR2(255 BYTE)	Yes	(null)	10	(null)
11	CONTACT	VARCHAR2(255 BYTE)	Yes	(null)	11	(null)
12	EVENTDATE	VARCHAR2(255 BYTE)	Yes	(null)	12	(null)
13	CHARTDATE	VARCHAR2(255 BYTE)	Yes	(null)	13	(null)

Create table: Hospital_Beds

```
CREATE TABLE "DB444"."HOSPITAL_BEDS"
(       "HOSPITAL_ID" NUMBER(38,0) NOT NULL ENABLE,
        "COUNTY_FIPS" NUMBER(38,0),
        "HOSPITAL_NAME" VARCHAR2(128 BYTE),
        "HOSPITAL_TYPE" VARCHAR2(128 BYTE),
        "NUM_LICENSED_BEDS" NUMBER(38,0),
        "NUM_STAFFED_BEDS" NUMBER(38,0),
        "NUM_ICU_BEDS" NUMBER(38,0),
        "ADULT_ICU_BEDS" NUMBER(38,0),
        "PEDI_ICE_BEDS" NUMBER(38,0),
        "BED_UTILIZATION" NUMBER(38,8),
        "POTENTIAL_INCR_IN_BED_CAPICITY" NUMBER(38,0),
```

Result:

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

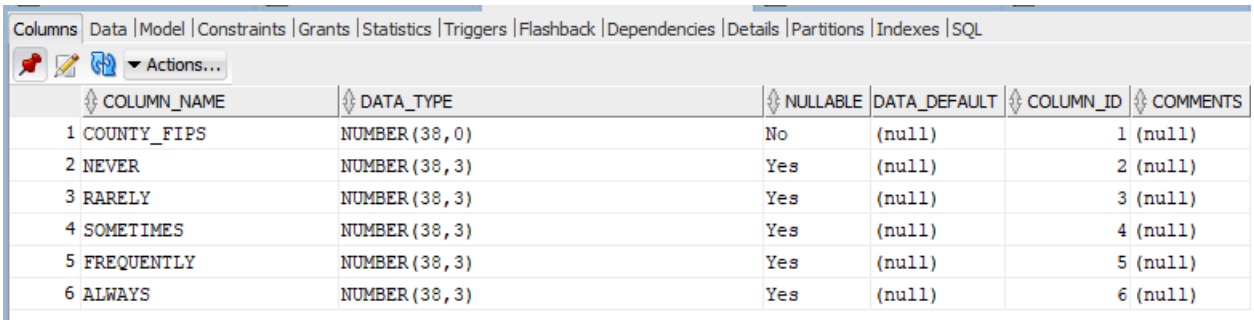
Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	HOSPITAL_ID	NUMBER(38,0)	No	(null)	1	(null)
2	COUNTY_FIPS	NUMBER(38,0)	Yes	(null)	2	(null)
3	HOSPITAL_NAME	VARCHAR2(128 BYTE)	Yes	(null)	3	(null)
4	HOSPITAL_TYPE	VARCHAR2(128 BYTE)	Yes	(null)	4	(null)
5	NUM_LICENSED_BEDS	NUMBER(38,0)	Yes	(null)	5	(null)
6	NUM_STAFFED_BEDS	NUMBER(38,0)	Yes	(null)	6	(null)
7	NUM_ICU_BEDS	NUMBER(38,0)	Yes	(null)	7	(null)
8	ADULT_ICU_BEDS	NUMBER(38,0)	Yes	(null)	8	(null)
9	PEDI_ICE_BEDS	NUMBER(38,0)	Yes	(null)	9	(null)
10	BED_UTILIZATION	NUMBER(38,8)	Yes	(null)	10	(null)
11	POTENTIAL_INCR_IN_B...	NUMBER(38,0)	Yes	(null)	11	(null)
12	AVG_VENTILATOR_USAGE	NUMBER(38,0)	Yes	(null)	12	(null)

Create table: Mask_Used_By_County

```
CREATE TABLE "DB444"."MASK_USED_BY_COUNTY"
(       "COUNTY_FIPS" NUMBER(38,0),
        "NEVER" NUMBER(38,3),
        "RARELY" NUMBER(38,3),
        "SOMETIMES" NUMBER(38,3),
        "FREQUENTLY" NUMBER(38,3),
        "ALWAYS" NUMBER(38,3)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
```

Result:



Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

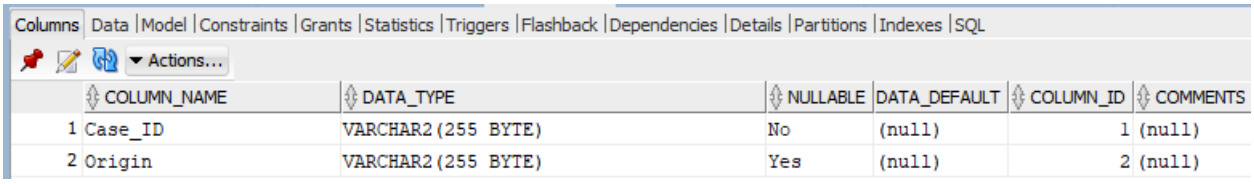
Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COUNTY_FIPS	NUMBER(38,0)	No	(null)	1	(null)
2	NEVER	NUMBER(38,3)	Yes	(null)	2	(null)
3	RARELY	NUMBER(38,3)	Yes	(null)	3	(null)
4	SOMETIMES	NUMBER(38,3)	Yes	(null)	4	(null)
5	FREQUENTLY	NUMBER(38,3)	Yes	(null)	5	(null)
6	ALWAYS	NUMBER(38,3)	Yes	(null)	6	(null)

Create Table: Origin

```
CREATE TABLE "DB444"."ORIGIN"
(       "Case_ID" VARCHAR2(255 BYTE),
        "Origin" VARCHAR2(255 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
```

Result:



Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

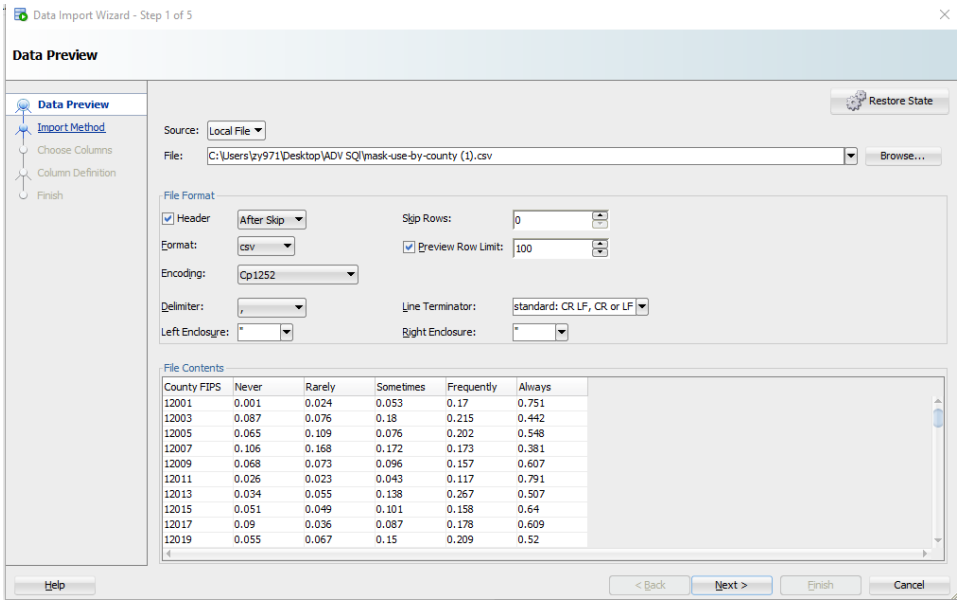
Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	Case_ID	VARCHAR2(255 BYTE)	No	(null)	1	(null)
2	Origin	VARCHAR2(255 BYTE)	Yes	(null)	2	(null)

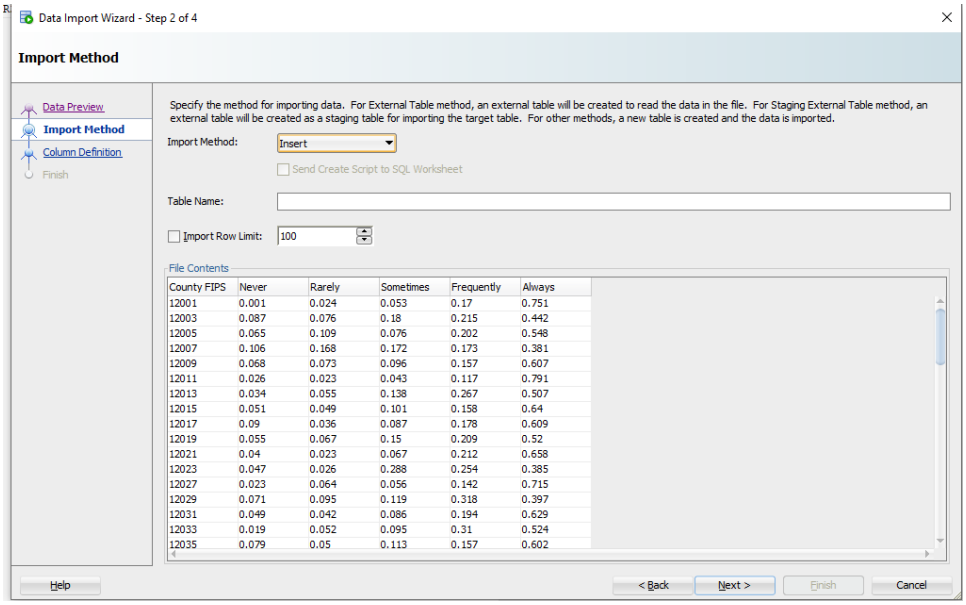
1.2.2 Data Loading

Since the data already exists, the only process we need to do is import the data source to the existing data table.

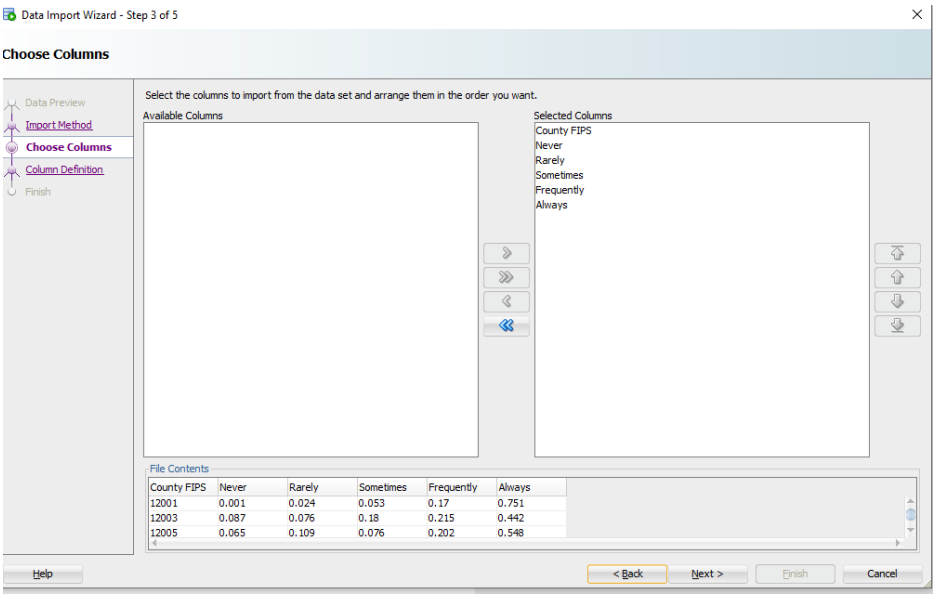
We use the import data function under the data table function.



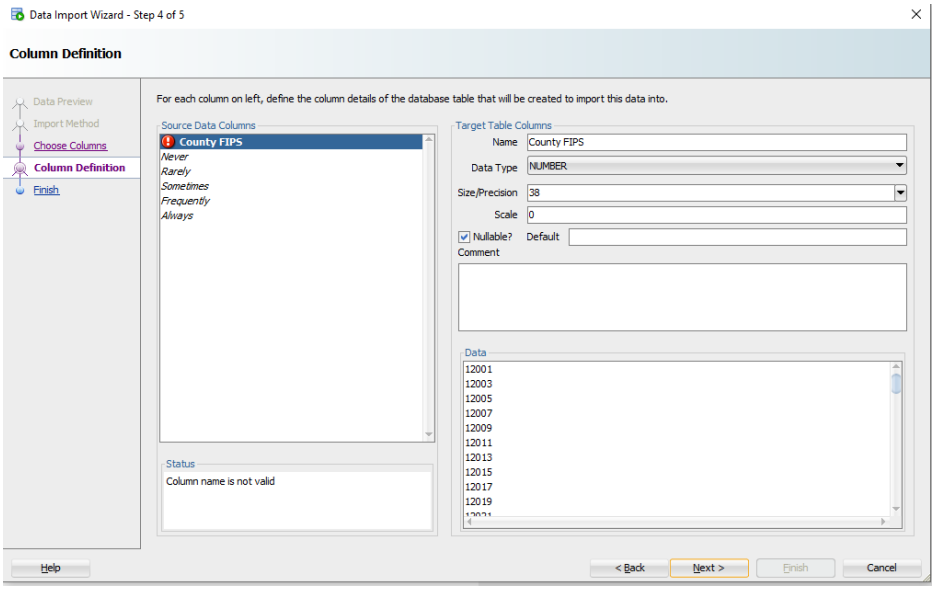
Adjust the Import Method and import the table name.



Select the columns that we want to be imported.



Define the table column based on our preset data dictionary.



The rest of the data loading is following the same procedures for the remaining tables.

Part 2: Query Writing

We are building some queries to test and validate the reliability of the database.

Query 1. Top 10 counties with most deaths/cases, for a simple and quick look at which counties have been hurt the most.

```
SELECT county_fips, county_name, casesall
FROM DB444.case_by_county
ORDER BY casesall DESC
FETCH FIRST 10 ROWS ONLY;
```

	COUNTY_FIPS	COUNTY_NAME	CASESALL
1	12025	Dade	170882
2	12011	Broward	77433
3	12099	Palm Beach	46698
4	12057	Hillsborough	42380
5	12095	Orange	40485
6	12031	Duval	30666
7	12103	Pinellas	22266
8	12071	Lee	20586
9	12105	Polk	20120
10	12021	Collier	12789

```
SELECT county_fips, county_name, deaths
FROM DB444.case_by_county
ORDER BY deaths DESC
FETCH FIRST 10 ROWS ONLY;
```

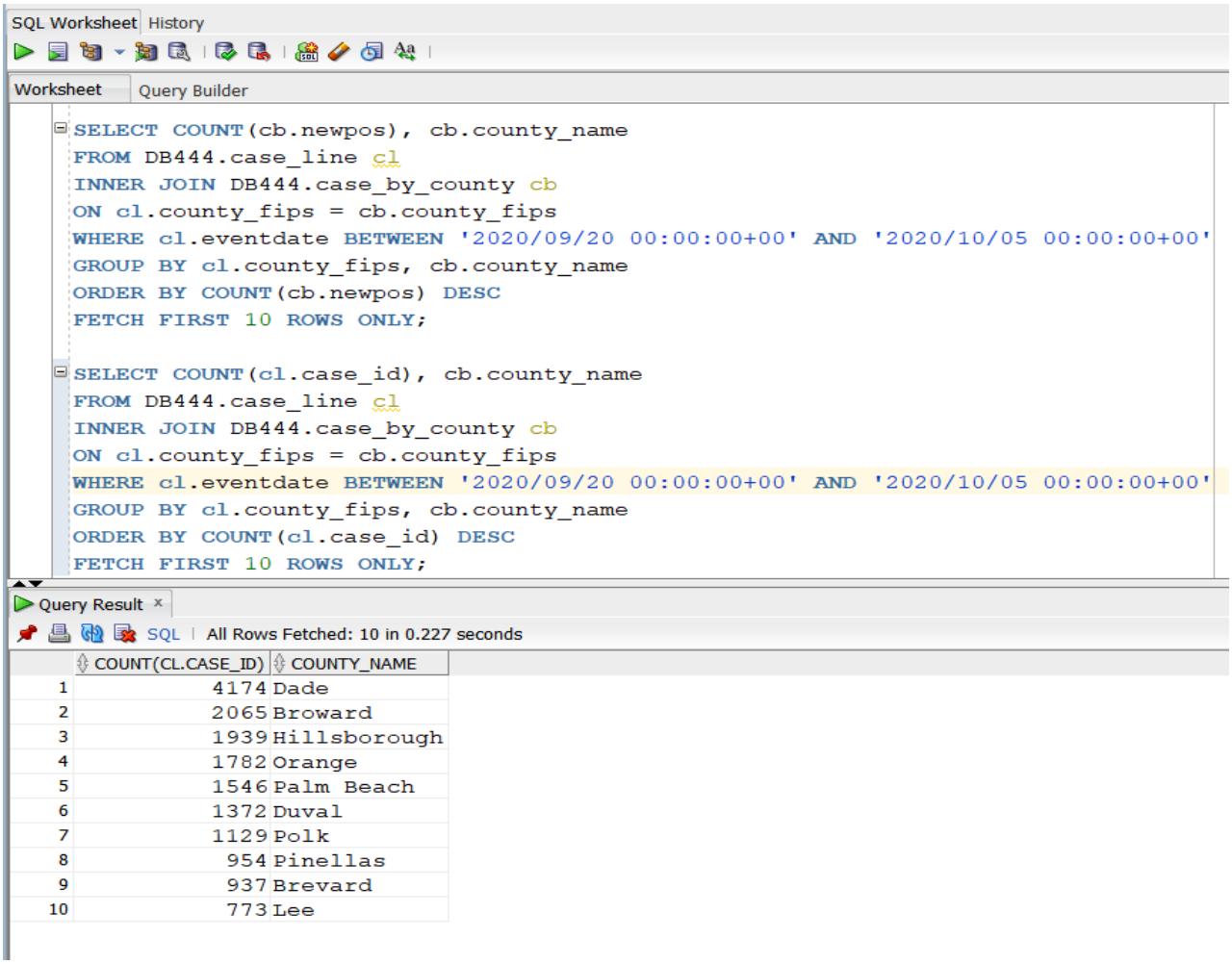
	COUNTY_FIPS	COUNTY_NAME	DEATHS
1	12025	Dade	3284
2	12011	Broward	1406
3	12099	Palm Beach	1383
4	12103	Pinellas	755
5	12057	Hillsborough	654
6	12105	Polk	531
7	12071	Lee	477
8	12095	Orange	470
9	12031	Duval	407
10	12009	Brevard	299

Query 2. Top 10 Covid-19 hotspots in the past 2 weeks (FL - counties)

a.
```
SELECT COUNT(cb.newpos), cb.county_name
FROM DB444.case_line cl
INNER JOIN DB444.case_by_county cb
ON cl.county_fips = cb.county_fips
WHERE cl.eventdate BETWEEN '2020/10/04 00:00:00+00' AND '2020/10/18 00:00:00+00'
GROUP BY cl.county_fips, cb.county_name
ORDER BY COUNT(cb.newpos) DESC
FETCH FIRST 10 ROWS ONLY;
```

b.
```
SELECT COUNT(cl.case_id), cb.county_name
FROM DB444.case_line cl
INNER JOIN DB444.case_by_county cb
ON cl.county_fips = cb.county_fips
WHERE cl.eventdate BETWEEN '2020/10/04 00:00:00+00' AND '2020/10/18 00:00:00+00'
GROUP BY cl.county_fips, cb.county_name
ORDER BY COUNT(cl.case_id) DESC
FETCH FIRST 10 ROWS ONLY;
```

*Both a & b queries resulted in same output



	COUNT(CL.CASE_ID)	COUNTY_NAME
1	4174	Dade
2	2065	Broward
3	1939	Hillsborough
4	1782	Orange
5	1546	Palm Beach
6	1372	Duval
7	1129	Polk
8	954	Pinellas
9	937	Brevard
10	773	Lee

Query 3. Comparing total number of beds versus the number of hospitalizations each county has experienced.

SELECT c.county_fips, county_name, (c.c_hospyes_nonres + c.c_hospyes_res) AS total_hospital, sum(num_licensed_beds) AS total_beds

FROM DB444.hospital_beds h INNER JOIN DB444.case_by_county c

ON h.county_fips = c.county_fips

GROUP BY c.county_fips, county_name, c.c_hospyes_nonres + c.c_hospyes_res

ORDER BY total_hospital DESC;

	COUNTY_FIPS	COUNTY_NAME	TOTAL_HOSPITAL	TOTAL_BEDS
1	12011	Broward	5938	6026
2	12099	Palm Beach	3687	4440
3	12103	Pinellas	2327	3874
4	12105	Polk	2116	1736
5	12057	Hillsborough	1885	4471
6	12071	Lee	1498	1614
7	12095	Orange	1411	3177
8	12031	Duval	1147	3456
9	12083	Marion	965	910
10	12021	Collier	929	633
11	12101	Pasco	843	1407
12	12033	Escambia	839	1648
13	12127	Volusia	837	1453

What I can see from this query is total hospitalizations in a county and how many total beds exist in that county's hospital. These numbers can't be used as a utilization percentage to see which counties have been overworked because utilization would be looking at how many cases that county has *at one time*, not in the total time frame like how it is in the cases_by_county table. I think there can still be some interpretation from this query, just seeing that some counties with few beds have had many hospitalizations, which may have caused lots of trouble for them. For example, Polk County has had the fourth most hospitalizations but only the ninth highest amount of beds. This could be a reason why Polk ranks as the county with the sixth most deaths. Meanwhile, Alachua county has the seventh-highest amount of hospital beds but only ranks 21st in hospitalizations. This could be a reason why Alachua has only had 61 deaths.

Query 4 (View best/worst mask usage in counties) either on always/sometimes results or a calculated value from all survey results.

```
SELECT
    county_fips,
    always,
    frequently,
    sometimes,
    rarely,
    never,
    ROUND(POWER((always+(frequently*0.75)+(sometimes*0.5)+(rarely*0.25)),5),3) as
mask_encounter
FROM mask_used_by_county
```

ORDER BY mask_encounter

FETCH FIRST 5 ROWS ONLY

	COUNTY_FIPS	ALWAYS	FREQUENTLY	SOMETIMES	RARELY	NEVER	MASK_ENCOUNTER
1	12079	0.34	0.264	0.148	0.085	0.164	0.102
2	12007	0.381	0.173	0.172	0.168	0.106	0.106
3	12047	0.365	0.227	0.235	0.071	0.101	0.136
4	12067	0.389	0.208	0.22	0.081	0.102	0.14
5	12121	0.369	0.227	0.244	0.055	0.105	0.14

Output is showing the 5 counties with the lowest mask encounter rate. Mask_encounter is a calculated rate, adopted from a study using similar mask information (source). For example, 0.102 means there is a 10.2% chance that everyone is masked in 5 random encounters. Add DESC to order and view counties with highest mask_encounter rate.

	COUNTY_FIPS	ALWAYS	FREQUENTLY	SOMETIMES	RARELY	NEVER	MASK_ENCOUNTER
1	12087	0.819	0.159	0.014	0.001	0.007	0.756
2	12001	0.751	0.17	0.053	0.024	0.001	0.627
3	12011	0.791	0.117	0.043	0.023	0.026	0.61
4	12099	0.784	0.116	0.05	0.02	0.03	0.594
5	12103	0.75	0.133	0.074	0.019	0.025	0.563

Query 5. To investigate the counties with the highest deaths, it could be important to look at the risk factors in that county.

It's been well documented that age is one of the biggest predictors of COVID severity. There are two age factors we selected to look at: the median age of people under investigation and the total of people under investigation ages 65 and older.

SELECT county_fips, county_name, puiagemedian, (age_65_74 + age_75_84 + age_85plus) AS At_Risk_Age, deaths

FROM DB444.case_by_county

ORDER BY deaths DESC;

	COUNTY_FIPS	COUNTY_NAME	PUIAGEMEDIAN	AT_RISK_AGE	DEATHS
1	12025	Dade	45	55096	3284
2	12011	Broward	45	32046	1406
3	12099	Palm Beach	49	28251	1383
4	12103	Pinellas	53	16605	755
5	12057	Hillsborough	43	17249	654
6	12105	Polk	45	8386	531
7	12071	Lee	50	12131	477
8	12095	Orange	43	16075	470
9	12031	Duval	44	14900	407
10	12009	Brevard	55	8387	299

Most of the highest death counties aren't very old, with only two of them over the age of 50, so there is likely a bigger predictor for those counties than age (likely population). There are still some counties that to take notice of in this age factor output. Sumter is number one here in PUI median age. Not only that,

Sumter is known for being home to The Villages, a large retirement community. In fact, Sumter has the highest average age of any county in the United States. Meanwhile the counties of Alachua, Hendry, Glades, Liberty, Hardee, and Leon rank near the bottom of the 67 counties. To compare the death rates of these counties:

SELECT county_fips, county_name, deaths, casesall, ROUND((deaths / casesall), 3) AS Death_Rate

FROM DB444.case_by_county

ORDER BY Death_Rate DESC;

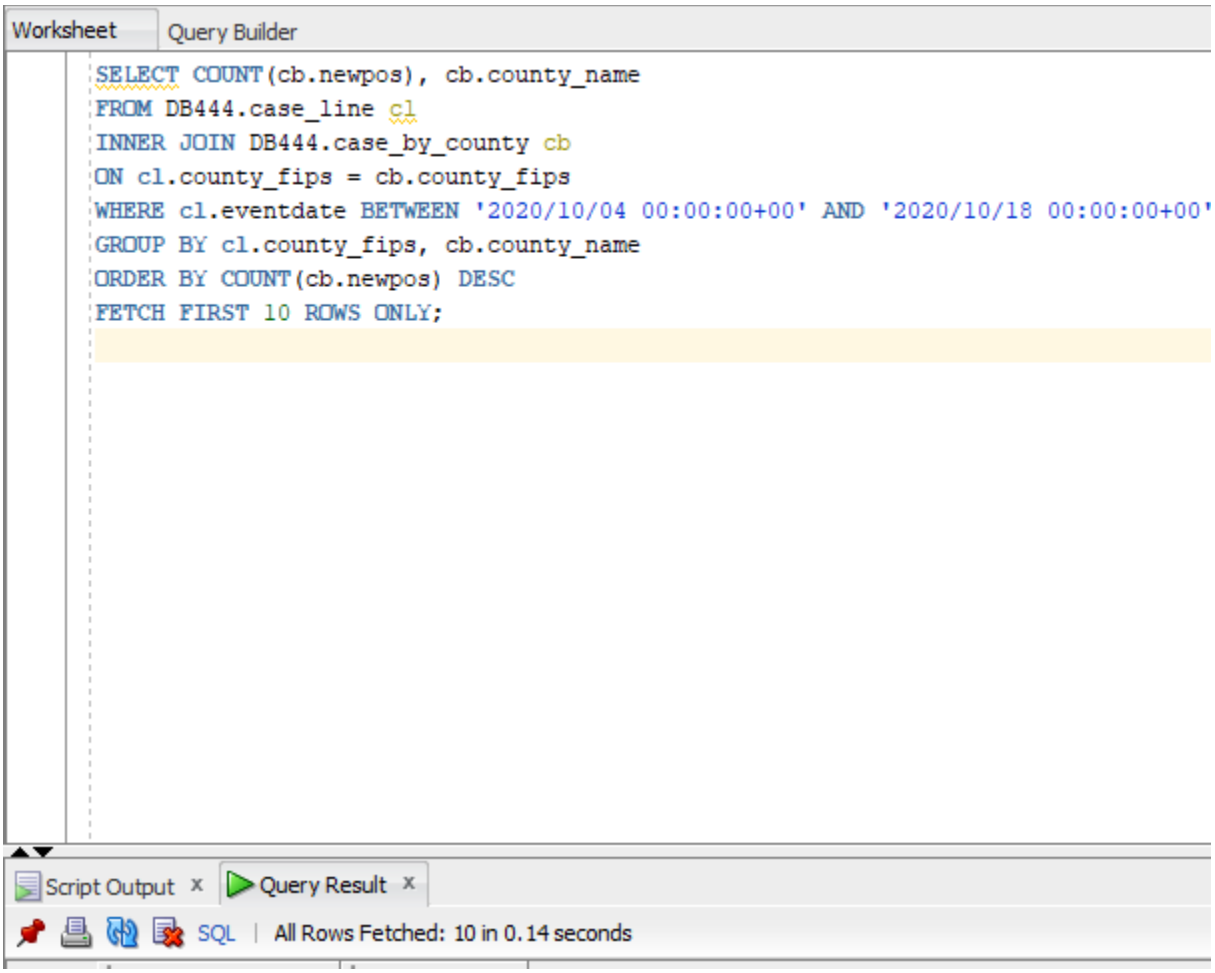
We see Sumter relatively high (tied for 12th in death rate), with Alachua, Glades, Hardee, and Leon all near the bottom:

	COUNTY_FIPS	COUNTY_NAME	DEATHS	CASESALL	DEATH_RATE	PUIAGEMEDIAN
1	12015	Charlotte	136	3097	0.044	58
2	12053	Hernando	132	3194	0.041	51
3	12055	Highlands	90	2186	0.041	55
4	12017	Citrus	107	2793	0.038	56
5	12061	Indian River	120	3279	0.037	55
6	12111	St. Lucie	291	8120	0.036	49
7	12103	Pinellas	755	22266	0.034	53
8	12009	Brevard	299	9154	0.033	55
9	12115	Sarasota	261	8233	0.032	55
10	12083	Marion	295	9806	0.03	51
11	12099	Palm Beach	1383	46698	0.03	49
12	12085	Martin	143	4942	0.029	51
13	12063	Jackson	81	2840	0.029	44
14	12119	Sumter	70	2397	0.029	66
62	12049	Hardee	11	1375	0.008	35
63	12007	Bradford	9	1135	0.008	48
64	12043	Glades	4	545	0.007	39
65	12001	Alachua	61	8352	0.007	39
66	12073	Leon	80	10710	0.007	35
67	12037	Franklin	4	628	0.006	45

Part 3: Performance Tuning

We are creating an index for a table to test the performance of the database.

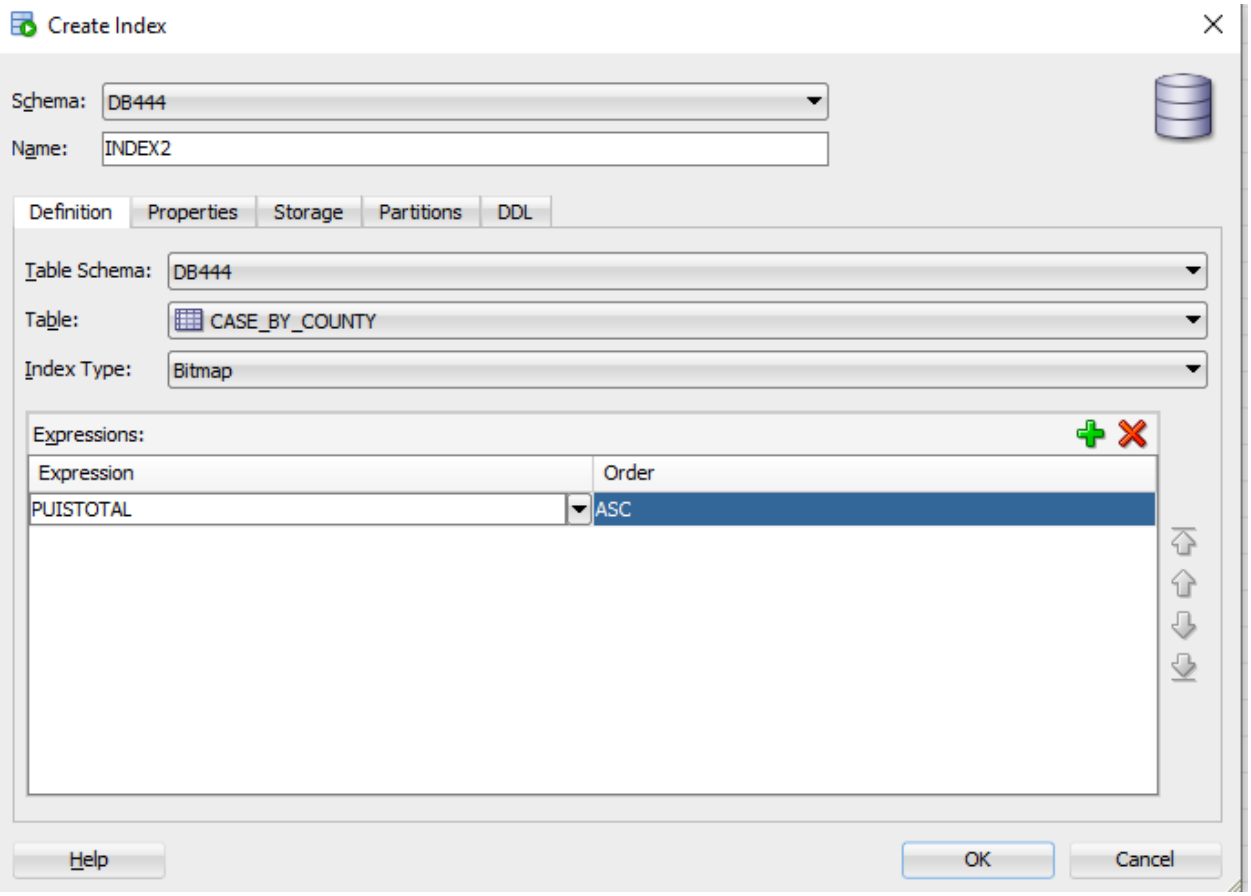
Taking our first Query test as an example. It took about 0.14 sec to finish the job.



SQL | All Rows Fetched: 10 in 0.14 seconds

	COUNT(CB.NEWPOS)	COUNTY_NAME
1	921	Dade
2	405	Hillsborough
3	390	Broward
4	387	Duval
5	385	Orange
6	261	Polk
7	250	Palm Beach
8	242	Lee
9	159	Pinellas
10	148	Leon

We add an index to the Case_by_county table.



Worksheet | Query Builder

```
SELECT COUNT(cb.newpos), cb.county_name
FROM DB444.case_line cl
INNER JOIN DB444.case_by_county cb
ON cl.county_fips = cb.county_fips
WHERE cl.eventdate BETWEEN '2020/10/04 00:00:00+00' AND '2020/10/18 00:00:00+00'
GROUP BY cl.county_fips, cb.county_name
ORDER BY COUNT(cb.newpos) DESC
FETCH FIRST 10 ROWS ONLY;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.086 seconds

	COUNT(CB.NEWPOS)	COUNTY_NAME
1	921	Dade
2	405	Hillsborough
3	390	Broward
4	387	Duval
5	385	Orange
6	261	Polk
7	250	Palm Beach
8	242	Lee
9	159	Pinellas
10	148	Leon

After the index process is done, the whole time reduced a little bit to around 0.086 sec.

Function-based Indexes

There is a use for function-based indexes in our database, as evidenced by some of the queries executed in Part 2. Specifically, there were two attributes we created as functions, mask-encounter rate and death rate.

There are queries in the future that we might execute that could call on these function-created attributes. We can compare the "cost" in execution plans on some experimental queries to see if building these indexes would improve the performance of our database.

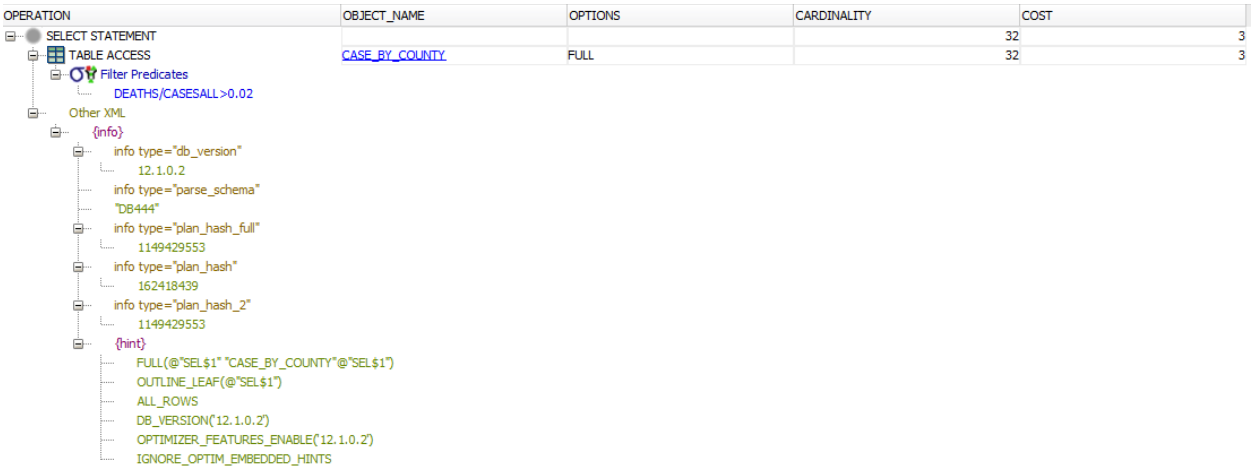
We can test a function-based index for death rate using the following query:

SELECT county_fips, county_name, ROUND((deaths / casesall), 3) AS Death_Rate

FROM case_by_county

WHERE (deaths / casesall) > 0.02;

The explain plan without first creating an index:



The explain plan for the same query after adding an index, created from CREATE INDEX DeathRate_Index ON case_by_county(deaths/casesall);

We see the cost reduce from (3+3) to (2+2+1). There is value in having this index.

Next we can test a query for mask-encounter rate.

SELECT

county_fips, ROUND(POWER((always+(frequently*0.75)+(sometimes*0.5)+(rarely*0.25)),5),3) as mask_encounter

FROM mask_used_by_county

WHERE POWER((always+(frequently*0.75)+(sometimes*0.5)+(rarely*0.25)),5) > 0.5;

The original explain plan:

CREATE INDEX Mask_rate_Index ON DB444.mask_used_by_County(POWER((always+(frequently*0.75)+(sometimes*0.5)+(rarely*0.25)),5));

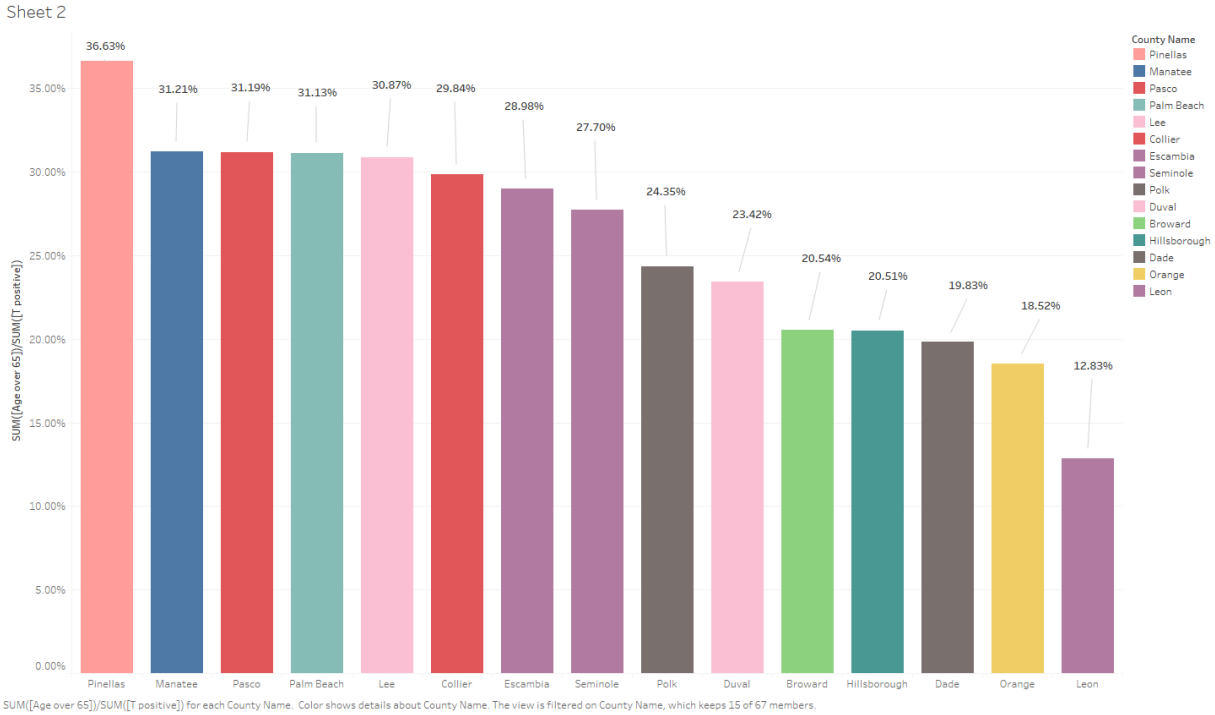
After we create that index and run the same query, the explain plan is:

Once again, we see a reduction in cost using the index (from 3+3 to 2+2+1). There is value in having this index.

Part 4: Other Topics

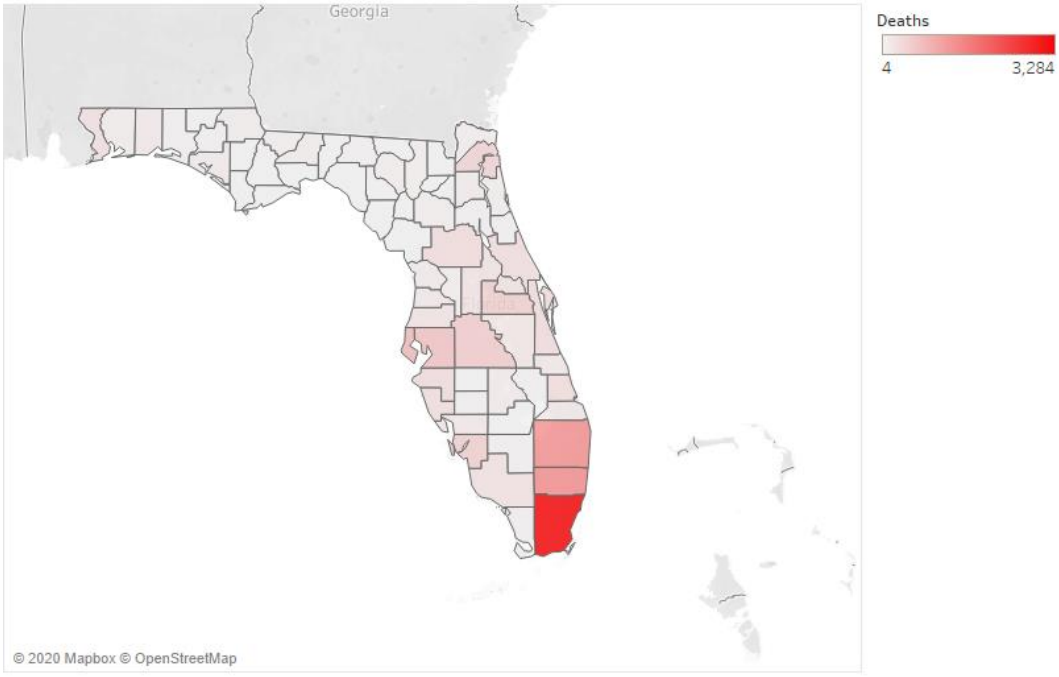
4.1 Data Visualization

Top 15 counties that have the highest positive rate in age group above 65.



Another visualization was made in Tableau, a map of Florida with county lines to show which counties were hit hardest. An interactive view of the chart is available here on Tableau Public. The chart has additional details of cases, death rates, and mask-encounter rate to give a quick generalization of each county. Once again, the mask-encounter rate is the probability of you seeing everyone being masked in five separate encounters. I believe this visualization is interesting because it stresses just how hard Miami-Dade county was hit. Certainly, some counties show as darker than others, but none sticks out more than Miami-Dade's color compared to the others.

Florida Counties by Death



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Deaths. Details are shown for State, County FIPS (mask-use-by-county) and County Name.