IT3031 - Database Systems and Data driven Applications

Assignment 1

IT17179386 – Jayasuriya K.E

Question 1

Report 1 -: If we can produce total confirmed, deaths, recovered for a country, it will useful to our leaders to decide how much resources assign to health ministry. In addition, Doctors and medical officers can identify how much attention they will have to give for all patients.

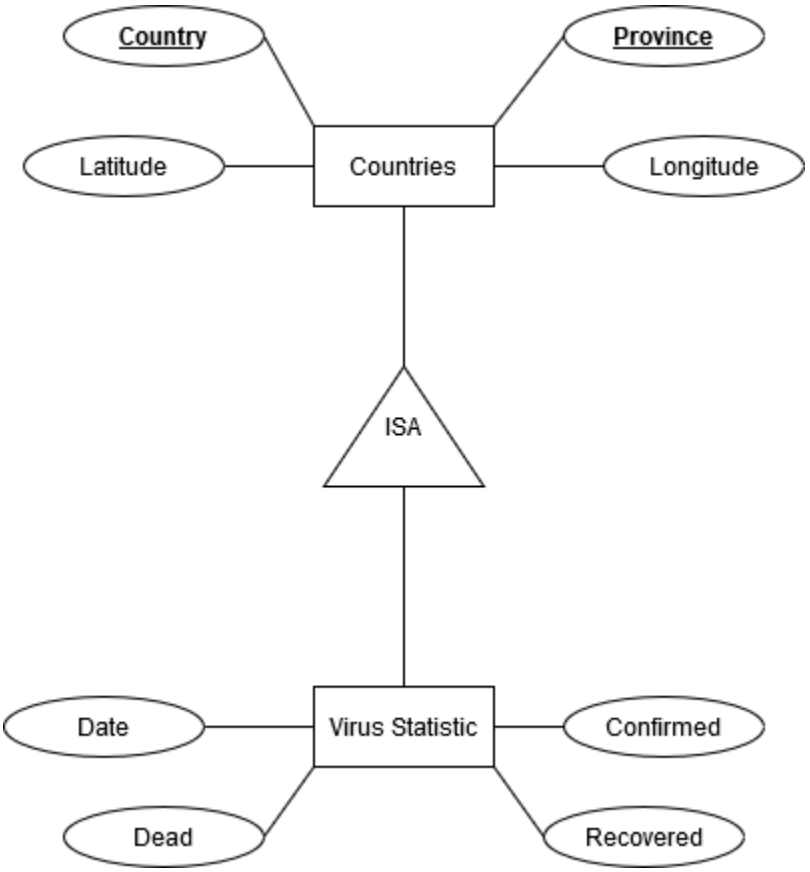
Report 2 -: If we can get total number of confirmed, deaths, recovered by province, it will useful to our leaders to decide which provinces should be lockdown and which provinces need more attention. In addition, Doctors and medical officers can decide how much medical resources need to handle this.

Report 3 -: If we can get total number of confirmed, deaths, recovered by all countries, it will useful to leaders to decide which countries need more help, compare their medical strength against other countries and also if any other country successfully prevent this virus our leaders can get information from that countries to prevent virus spread in our country.

Report 4 -: By getting death ratio and recovered ratio, Scientists can guess how many deaths will be next round. There for Leaders, medical staff can ready for upcoming patients.

Report 5 -: By getting remaining patients count and percentage in hospital and percentage of dead and recovered patient's, medical analysists can create chart of patient percentage and comparing it, they can decide our country at how much risk and inform the leaders to take proper actions

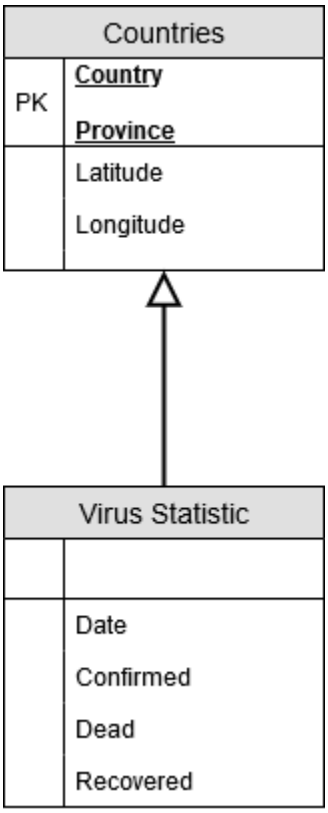
Question 2



I draw this EER because every country has particular virus statistic. Each virus statistic report represents their country. There for every report has unique qualities for each country. There for I have decide ISA relationship is suitable for this.

I have use **Country** and **Province** as **Composite keys** because if get these columns as individual we can't call it as primary key because province has nulls and country has duplicates but if we get these two columns together there are no nulls and duplicates. There for I have decided to use them both as composites.

Question 3

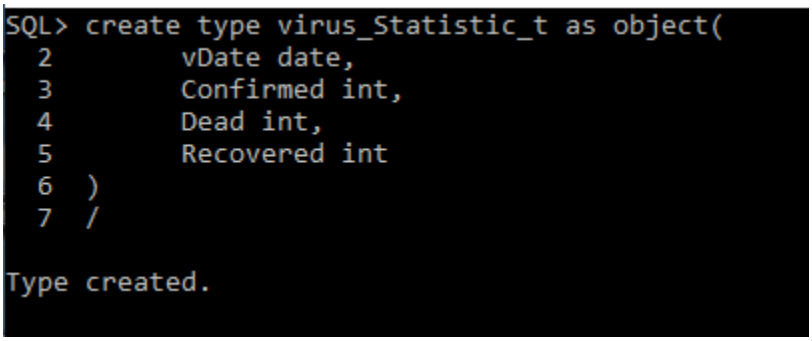


I have use **Country** and **Province** as **Composite keys** because if get these columns as individual we can't call it as primary key because province has nulls and country has duplicates but if we get these two columns together there are no nulls and duplicates. There for I have decided to use them both as composites.

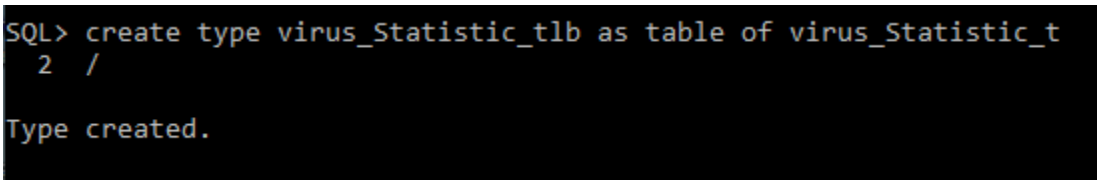
Question 4

By referring EER diagram and Object Relational Data Model, I have created these oracle scripts for creating my table and nested tables

```
create type virus_Statistic_t as object(
    vDate date,
    Confirmed int,
    Dead int,
    Recovered int
)
/
```



```
create type virus_Statistic_tlb as table of virus_Statistic_t
/
```



```
create type countries_t as object(
   Province varchar2(50),
   Country varchar2(100),
   Latitude Number,
   Longitude Number,
   virus virus_Statistic_tlb
)
/
```

```
SQL> create type countries_t as object(
  2      Province varchar2(50),
  3      Country varchar2(100),
  4      Latitude Number,
  5      Longitude Number,
  6      virus virus_Statistic_tlb
  7  )
  8  /

Type created.
```

```
create table countries of countries_t (
      Province null,
      Country not null,
      Latitude not null,
      Longitude not null,
      primary key(Province, Country)
) nested table virus store as virus_ntb;
```

```
SQL> create table countries of countries_t (
  2  Province null,
  3         Country not null,
  4         Latitude not null,
  5         Longitude not null,
  6          primary key(Province, Country)
  7  ) nested table virus store as virus_ntb;

Table created.
```

Question 5

Before insert data into database I have investigate given three csv files. I have realize there are lot of date type columns. Then I have decided to convert these columns into one columns

Problems -: my **#** marked date columns display like' 2002-dd-yy' but when I asked my friends they told me they don't have this kind of issue. I cannot change the column name also. It looks like locked columns. Then I investigate this problem and I realized I am using different version of excel than my friends. I think this is the problem. I have asked my friends to send their three-csv files but no luck. I am out of resources to download new office package. I have add that screenshot also

M	N	O	P	Q
1/30/20	1/31/20	2002-01-20	2002-02-20	2002-03-20
14	19	19	19	19
11	15	20	20	20

After I inserted values to the database, these dates are still same.

I have implement python code to convert these incrementing date columns into one column.

1. First I have loaded **time_series_19-covid-Confirmed.csv**

   import pandas as pd
   df = pd.read_csv (r'E:\time_series_19-covid-Confirmed.csv')
   df.head()

```
In [1]: import pandas as pd
        df = pd.read_csv (r'E:\time_series_19-covid-Confirmed.csv')
        df.head()
```

Out[1]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/12/20	3/13/20	3/14/20	3/15/20	3/16/20	3/17/20
0	NaN	Thailand	15.0000	101.0000	2	3	5	7	8	8	...	70	75	82	114	147	177
1	NaN	Japan	36.0000	138.0000	2	1	2	2	4	4	...	639	701	773	839	825	878
2	NaN	Singapore	1.2833	103.8333	0	1	3	3	4	5	...	178	200	212	226	243	266
3	NaN	Nepal	28.1667	84.2500	0	0	0	1	1	1	...	1	1	1	1	1	1
4	NaN	Malaysia	2.5000	112.5000	0	0	0	3	4	4	...	149	197	238	428	566	673

5 rows × 64 columns

2. Then I have removed all Nan values. (It is not compulsory but I have done it for good)

```
import numpy as np
df1 = df.replace(np.nan, '', regex=True)
df1.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/12/20	3/13/20	3/14/20	3/15/20	3/16/20	3/17/20
0		Thailand	15.0000	101.0000	2	3	5	7	8	8	...	70	75	82	114	147	177
1		Japan	36.0000	138.0000	2	1	2	2	4	4	...	639	701	773	839	825	878
2		Singapore	1.2833	103.8333	0	1	3	3	4	5	...	178	200	212	226	243	266
3		Nepal	28.1667	84.2500	0	0	0	1	1	1	...	1	1	1	1	1	1
4		Malaysia	2.5000	112.5000	0	0	0	3	4	4	...	149	197	238	428	566	673

5 rows × 64 columns

3. Then I have convert date columns and insert into one column with infected count

```
df2 = df.melt(id_vars=["Province/State", "Country/Region", "Lat", "Long"], var_name="Date", value_name="Confirmed")

df2 = df2.fillna(method='ffill')
df3 = df2.groupby(['Province/State', 'Country/Region', 'Lat', 'Long', 'Date', 'Confirmed']).sum()

df3.head()
```

Province/State	Country/Region	Lat	Long	Date	Confirmed
Adams, IN	US	39.8522	-77.2865	1/22/20	0
				1/23/20	0
				1/24/20	0
				1/25/20	0
				1/26/20	0

4. Then I have export that dataset into new csv. **You can see column names repeated in csv but I have fixed that problem in oracle**

   df3.to_csv('E:\Confirmed.csv')

```
df3.to_csv('E:\Confirmed.csv')
```

	A	B	C	D	E	F
	Province/	Country/R	Lat	Long	Date	Confirmed
2	Adams, IN	US	39.8522	-77.2865	1/22/20	0
3	Adams, IN	US	39.8522	-77.2865	1/23/20	0
4	Adams, IN	US	39.8522	-77.2865	1/24/20	0
5	Adams, IN	US	39.8522	-77.2865	1/25/20	0
6	Adams, IN	US	39.8522	-77.2865	1/26/20	0
7	Adams, IN	US	39.8522	-77.2865	1/27/20	0
8	Adams, IN	US	39.8522	-77.2865	1/28/20	0
9	Adams, IN	US	39.8522	-77.2865	1/29/20	0
10	Adams, IN	US	39.8522	-77.2865	1/30/20	0
11	Adams, IN	US	39.8522	-77.2865	1/31/20	0
12	Adams, IN	US	39.8522	-77.2865	########	0

5. I have applied this python code to other two csv files also and get these converted csv files.

	A	B	C	D	E	F
1	Province/	Country/R	Lat	Long	Date	Dead
2	Adams, IN	US	39.8522	-77.2865	1/22/20	0
3	Adams, IN	US	39.8522	-77.2865	1/23/20	0
4	Adams, IN	US	39.8522	-77.2865	1/24/20	0
5	Adams, IN	US	39.8522	-77.2865	1/25/20	0
6	Adams, IN	US	39.8522	-77.2865	1/26/20	0
7	Adams, IN	US	39.8522	-77.2865	1/27/20	0
8	Adams, IN	US	39.8522	-77.2865	1/28/20	0
9	Adams, IN	US	39.8522	-77.2865	1/29/20	0
10	Adams, IN	US	39.8522	-77.2865	1/30/20	0
11	Adams, IN	US	39.8522	-77.2865	1/31/20	0
12	Adams, IN	US	39.8522	-77.2865	########	0
13	Adams, IN	US	39.8522	-77.2865	########	0
14	Adams, IN	US	39.8522	-77.2865	########	0

	A	B	C	D	E	F
1	Province/	Country/R	Lat	Long	Date	Recovered
2	Adams, IN	US	39.8522	-77.2865	1/22/20	0
3	Adams, IN	US	39.8522	-77.2865	1/23/20	0
4	Adams, IN	US	39.8522	-77.2865	1/24/20	0
5	Adams, IN	US	39.8522	-77.2865	1/25/20	0
6	Adams, IN	US	39.8522	-77.2865	1/26/20	0
7	Adams, IN	US	39.8522	-77.2865	1/27/20	0
8	Adams, IN	US	39.8522	-77.2865	1/28/20	0
9	Adams, IN	US	39.8522	-77.2865	1/29/20	0
10	Adams, IN	US	39.8522	-77.2865	1/30/20	0
11	Adams, IN	US	39.8522	-77.2865	1/31/20	0
12	Adams, IN	US	39.8522	-77.2865	########	0
13	Adams, IN	US	39.8522	-77.2865	########	0
14	Adams, IN	US	39.8522	-77.2865	########	0
15	Adams, IN	US	39.8522	-77.2865	########	0
16	Adams, IN	US	39.8522	-77.2865	2/13/20	0

6. After those steps I have get three csv files named **Confirmed.csv, Dead.csv, Recovered.csv.** Then I have implemented another python code to merge these three files and produce one csv.

```
import pandas as pd
import numpy as np
import glob

confirmedCsv = pd.DataFrame()
for f in glob.glob("Confirmed.csv"):
   df = pd.read_csv(f)
   finalCsv = confirmedCsv.append(df,ignore_index=True)

finalCsv=finalCsv.replace(np.nan,'',regex=True)
finalCsv.head()
```

```
In [1]: import pandas as pd
        import numpy as np
```

```
In [2]: import glob
        confirmedCsv = pd.DataFrame()
        for f in glob.glob("Confirmed.csv"):
            df = pd.read_csv(f)
            finalCsv = confirmedCsv.append(df,ignore_index=True)
```

```
In [3]: finalCsv=finalCsv.replace(np.nan,'',regex=True)
        finalCsv.head()
```

Out[3]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed
0	Adams, IN	US	39.8522	-77.2865	1/22/20	0
1	Adams, IN	US	39.8522	-77.2865	1/23/20	0
2	Adams, IN	US	39.8522	-77.2865	1/24/20	0
3	Adams, IN	US	39.8522	-77.2865	1/25/20	0
4	Adams, IN	US	39.8522	-77.2865	1/26/20	0

```
deadCsv = pd.read_csv("Dead.csv")

deadCsv=deadCsv.replace(np.nan,'',regex=True)

deadCsv.head()


finalCsv = pd.merge(finalCsv, deadCsv , how='left')

finalCsv.head()
```

In [4]:
```
deadCsv = pd.read_csv("Dead.csv")
deadCsv=deadCsv.replace(np.nan,'',regex=True)
deadCsv.head()
```

Out[4]:

	Province/State	Country/Region	Lat	Long	Date	Dead
0	Adams, IN	US	39.8522	-77.2865	1/22/20	0
1	Adams, IN	US	39.8522	-77.2865	1/23/20	0
2	Adams, IN	US	39.8522	-77.2865	1/24/20	0
3	Adams, IN	US	39.8522	-77.2865	1/25/20	0
4	Adams, IN	US	39.8522	-77.2865	1/26/20	0

In [5]:
```
finalCsv = pd.merge(finalCsv, deadCsv , how='left')
finalCsv.head()
```

Out[5]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Dead
0	Adams, IN	US	39.8522	-77.2865	1/22/20	0	0
1	Adams, IN	US	39.8522	-77.2865	1/23/20	0	0
2	Adams, IN	US	39.8522	-77.2865	1/24/20	0	0
3	Adams, IN	US	39.8522	-77.2865	1/25/20	0	0

---

recoveredCsv = pd.read_csv("Recovered.csv")

recoveredCsv=recoveredCsv.replace(np.nan,'',regex=True)

recoveredCsv.head()

In [6]:
```
recoveredCsv = pd.read_csv("Recovered.csv")
recoveredCsv=recoveredCsv.replace(np.nan,'',regex=True)
recoveredCsv.head()
```

Out[6]:

	Province/State	Country/Region	Lat	Long	Date	Recovered
0	Adams, IN	US	39.8522	-77.2865	1/22/20	0
1	Adams, IN	US	39.8522	-77.2865	1/23/20	0
2	Adams, IN	US	39.8522	-77.2865	1/24/20	0
3	Adams, IN	US	39.8522	-77.2865	1/25/20	0
4	Adams, IN	US	39.8522	-77.2865	1/26/20	0

finalCsv = pd.merge(finalCsv, recoveredCsv , how='left')

finalCsv=finalCsv.groupby(['Province/State','Country/Region','Lat','Long','Date', 'Confirmed', 'Dead', 'Recovered']).sum()

finalCsv.head()

finalCsv.to_csv('final.csv')

In [7]:
```
finalCsv = pd.merge(finalCsv, recoveredCsv , how='left')
finalCsv=finalCsv.groupby(['Province/State','Country/Region','Lat','Long','Date', 'Confirmed', 'Dead', 'Recovered']).sum()
finalCsv.head()
```

Out[7]:

Province/State	Country/Region	Lat	Long	Date	Confirmed	Dead	Recovered
Adams, IN	US	39.8522	-77.2865	1/22/20	0	0	0.0
				1/23/20	0	0	0.0
				1/24/20	0	0	0.0
				1/25/20	0	0	0.0
				1/26/20	0	0	0.0

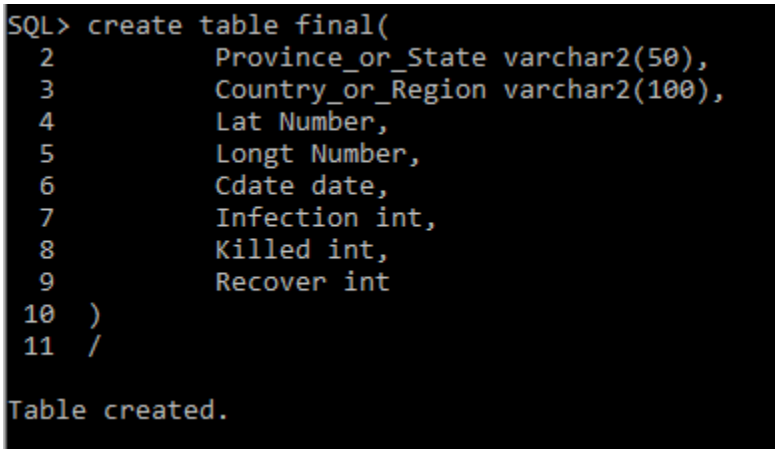
In [8]:
```
finalCsv.to_csv('final.csv')
```

	A	B	C	D	E	F	G	H
1	Province/	Country/R	Lat	Long	Date	Confirmed	Dead	Recovered
2	Adams, IN	US	39.8522	-77.2865	1/22/20	0	0	0
3	Adams, IN	US	39.8522	-77.2865	1/23/20	0	0	0
4	Adams, IN	US	39.8522	-77.2865	1/24/20	0	0	0
5	Adams, IN	US	39.8522	-77.2865	1/25/20	0	0	0
6	Adams, IN	US	39.8522	-77.2865	1/26/20	0	0	0
7	Adams, IN	US	39.8522	-77.2865	1/27/20	0	0	0
8	Adams, IN	US	39.8522	-77.2865	1/28/20	0	0	0
9	Adams, IN	US	39.8522	-77.2865	1/29/20	0	0	0
10	Adams, IN	US	39.8522	-77.2865	1/30/20	0	0	0
11	Adams, IN	US	39.8522	-77.2865	1/31/20	0	0	0
12	Adams, IN	US	39.8522	-77.2865	#########	0	0	0
13	Adams, IN	US	39.8522	-77.2865	#########	0	0	0
14	Adams, IN	US	39.8522	-77.2865	#########	0	0	0

Above Image, represent final csv file after merging 3 csv files into one. In final.csv, files have 28856 rows likewise other csv files.

7. After all those steps, now we have to insert final.csv data into our created oracle schema. There for I have decided to create new table called final and load my csv data into that table and after that passing data to my oracle schema from final table.

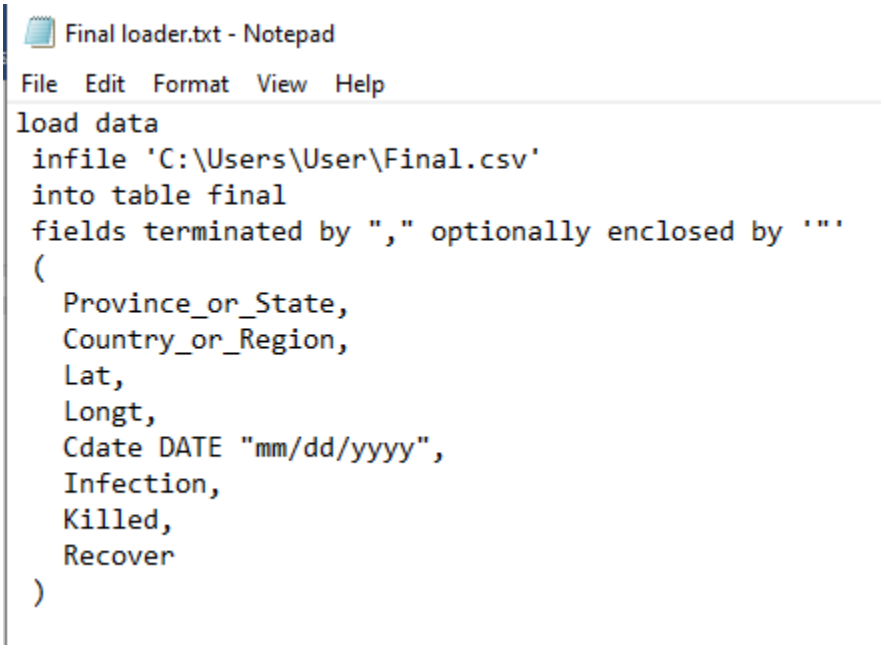
```
create table final(
        Province_or_State varchar2(50),
        Country_or_Region varchar2(100),
        Lat Number,
        Longt Number,
        Cdate date,
        Infection int,
        Killed int,
        Recover int
)
/
```



8. Now the real challenge is occurring, that is load csv data into final table. After some studying I have found by using **SQL Loader,** we can insert csv data into that table. There for I have created text file named **Final loader.txt.** After that I have add configurations to that text file to store and pass csv data to final table

- **Create Final loader.txt and add these configurations**

```
load data
 infile 'C:\Users\User\Final.csv'
 into table final
 fields terminated by "," optionally enclosed by '"'
 (
        Province_or_State,
        Country_or_Region,
        Lat,
        Longt,
        Cdate DATE "mm/dd/yyyy",
        Infection,
        Killed,
        Recover
 )
```



- Then we have to run **SQL Loader** in our cmd by running this code -: **sqlldr username/password**
- After that cmd display **"control ="** and we have to give our **Final loader.txt** location to run that text file with running configuration.

- First, we need to exit **SQL Plus** to run **SQL Loader.**

```
SQL> create table final(
  2          Province_or_State varchar2(50),
  3          Country_or_Region varchar2(100),
  4          Lat Number,
  5          Longt Number,
  6          Cdate date,
  7          Infection int,
  8          Killed int,
  9          Recover int
 10  )
 11  /

Table created.

SQL> exit
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

C:\Users\User>sqlldr dsda/dsda

control = C:\Users\User\Desktop\Final loader.txt
```

After entering controlling values, we can see our csv data (28856 rows) inserted into final table.

```
Commit point reached - logical record count 27147
Commit point reached - logical record count 27211
Commit point reached - logical record count 27275
Commit point reached - logical record count 27339
Commit point reached - logical record count 27403
Commit point reached - logical record count 27467
Commit point reached - logical record count 27531
Commit point reached - logical record count 27595
Commit point reached - logical record count 27659
Commit point reached - logical record count 27723
Commit point reached - logical record count 27787
Commit point reached - logical record count 27851
Commit point reached - logical record count 27915
Commit point reached - logical record count 27979
Commit point reached - logical record count 28043
Commit point reached - logical record count 28107
Commit point reached - logical record count 28171
Commit point reached - logical record count 28235
Commit point reached - logical record count 28299
Commit point reached - logical record count 28363
Commit point reached - logical record count 28427
Commit point reached - logical record count 28491
Commit point reached - logical record count 28555
Commit point reached - logical record count 28619
Commit point reached - logical record count 28683
Commit point reached - logical record count 28747
Commit point reached - logical record count 28811
Commit point reached - logical record count 28856

C:\Users\User>
```

9. Now we have to pass all values from final table to our oracle schema. I have use this oracle script for insert values to main table and nested table. After Inserting there are 481 rows was created. **And also we saw duplicate Country and province rows in final.csv. There for I have used group by option to arrange duplicates into unique. Now again we have to log into sqlplus because of sqlloader**

```
INSERT INTO countries(Province, Country, Latitude, Longitude, virus)
SELECT  Province_or_State, Country_or_Region, Lat, Longt,
    CAST(MULTISET(SELECT virus_Statistic_t(Cdate, Infection, Killed, Recover)
            FROM   final f2
            WHERE  f2.Province_or_State = f1.Province_or_State
            AND    f2.Country_or_Region = f1.Country_or_Region
            AND    f2.Lat = f1.Lat
            AND    f2.Longt = f1.Longt
            ) AS virus_Statistic_tlb
    )
FROM final f1
GROUP BY Province_or_State, Country_or_Region, Lat, Longt
/
```

```
SQL> INSERT INTO countries(Province, Country, Latitude, Longitude, virus)
  2  SELECT  Province_or_State, Country_or_Region, Lat, Longt,
  3          CAST(MULTISET(SELECT virus_Statistic_t(Cdate, Infection, Killed, Recover)
  4                        FROM   final f2
  5                        WHERE  f2.Province_or_State = f1.Province_or_State
  6                        AND    f2.Country_or_Region = f1.Country_or_Region
  7                        AND    f2.Lat = f1.Lat
  8                        AND    f2.Longt = f1.Longt
  9                        ) AS virus_Statistic_tlb
 10          )
 11  FROM final f1
 12  GROUP BY Province_or_State, Country_or_Region, Lat, Longt
 13  /

481 rows created.

SQL>
```

10. **After adding that we have to type this command to commit that records to our countries table unless if you close cmd and open it again and when you try to select data from countries table there will be no data in that table. There for after passing values one table to another table make sure that values are committed.**

```
Commit

/
```

11. **You can see 481 rows created in main table and nested table have lot of rows. Let's see select first row of countries table. I have added sample of first raw.**

```
column Province format a10

column Country format a10

column Latitude format 999999999

column Longitude format 999999999

column virus format a30

select * from countries where rownum= 1

/
```

```
SQL> column Province format a10
SQL> column Country format a10
SQL> column Latitude format 999999999
SQL> column Longitude format 999999999
SQL> column virus format a30
SQL> select * from countries where rownum= 1
  2  /

PROVINCE   COUNTRY      LATITUDE  LONGITUDE VIRUS(VDATE, CONFIRMED, DEAD,
---------- ---------- ---------- ---------- ------------------------------
Alameda Co US                 38       -122 VIRUS_STATISTIC_TLB(VIRUS_STAT
unty, CA                                    ISTIC_T('22-JAN-20', 0, 0, 0),
                                             VIRUS_STATISTIC_T('23-JAN-20'
                                            , 0, 0, 0), VIRUS_STATISTIC_T(
                                            '24-JAN-20', 0, 0, 0), VIRUS_S
                                            TATISTIC_T('25-JAN-20', 0, 0,
                                            0), VIRUS_STATISTIC_T('26-JAN-
                                            20', 0, 0, 0), VIRUS_STATISTIC
                                            _T('27-JAN-20', 0, 0, 0), VIRU
                                            S_STATISTIC_T('28-JAN-20', 0,
                                            0, 0), VIRUS_STATISTIC_T('29-J

PROVINCE   COUNTRY      LATITUDE  LONGITUDE VIRUS(VDATE, CONFIRMED, DEAD,
---------- ---------- ---------- ---------- ------------------------------
                                            AN-20', 0, 0, 0), VIRUS_STATIS
                                            TIC_T('30-JAN-20', 0, 0, 0), V
                                            IRUS_STATISTIC_T('31-JAN-20',
                                            0, 0, 0), VIRUS_STATISTIC_T('0
                                            1-FEB-20', 0, 0, 0), VIRUS_STA
                                            TISTIC_T('10-FEB-20', 0, 0, 0)
                                            , VIRUS_STATISTIC_T('11-FEB-20
                                            ', 0, 0, 0), VIRUS_STATISTIC_T
                                            ('12-FEB-20', 0, 0, 0), VIRUS_
                                            STATISTIC_T('13-FEB-20', 0, 0,
                                             0), VIRUS_STATISTIC_T('14-FEB
```

Above screenshot represents sample of my first row

Question 6

I have implemented three member functions, which will help me to create five reports. I have implemented these member functions

- totalInfected -: to get total number of infected.
- totalDead -: to get total number of dead.
- totalRecovered -: to get total number of recovered.

I have not created function, which is implement time because I have previously told my csv files have junk date columns. I could not drop that column because others have it correctly.

ALTER TYPE countries_t ADD MEMBER FUNCTION totalInfected RETURN NUMBER CASCADE

/

ALTER TYPE countries_t ADD MEMBER FUNCTION totalDead RETURN NUMBER CASCADE

/

ALTER TYPE countries_t ADD MEMBER FUNCTION totalRecovered RETURN NUMBER CASCADE

/

```
SQL> ALTER TYPE countries_t ADD MEMBER FUNCTION totalInfected RETURN NUMBER CASCADE
  2  /

Type altered.

SQL> ALTER TYPE countries_t ADD MEMBER FUNCTION totalDead RETURN NUMBER CASCADE
  2  /

Type altered.

SQL> ALTER TYPE countries_t ADD MEMBER FUNCTION totalRecovered RETURN NUMBER CASCADE
  2  /

Type altered.

SQL>
```

Then I have created my functions

```
CREATE OR REPLACE
TYPE BODY countries_t AS
MEMBER FUNCTION totalInfected RETURN NUMBER IS
confirmed number;
BEGIN
    SELECT max(v.Confirmed)
    INTO confirmed
    FROM table(self.virus) v;
    RETURN confirmed;
END;
MEMBER FUNCTION totalDead RETURN NUMBER IS
dead number;
BEGIN
    SELECT max(v.Dead)
    INTO dead
    FROM table(self.virus) v;
    RETURN dead;
END;
MEMBER FUNCTION totalRecovered RETURN NUMBER IS
recover number;
BEGIN
    SELECT max(v.Recovered)
    INTO recover
    FROM table(self.virus) v;
    RETURN recover;
END;
END;
/
```

```
SQL> CREATE OR REPLACE
  2  TYPE BODY countries_t AS
  3  MEMBER FUNCTION totalInfected RETURN NUMBER IS
  4  confirmed number;
  5  BEGIN
  6          SELECT max(v.Confirmed)
  7          INTO confirmed
  8          FROM table(self.virus) v;
  9          RETURN confirmed;
 10  END;
 11  MEMBER FUNCTION totalDead RETURN NUMBER IS
 12  dead number;
 13  BEGIN
 14          SELECT max(v.Dead)
 15          INTO dead
 16          FROM table(self.virus) v;
 17          RETURN dead;
 18  END;
 19  MEMBER FUNCTION totalRecovered RETURN NUMBER IS
 20  recover number;
 21  BEGIN
 22          SELECT max(v.Recovered)
 23          INTO recover
 24          FROM table(self.virus) v;
 25          RETURN recover;
 26  END;
 27  END;
 28  /

Type body created.

SQL>
```

I have used **max()** function to get total confirmed, dead and recovered because in the csv total of these columns situated in last date. There for I have get **max()** instead of **sum()** function.

Question 7

By using three functions, which are created in Question 6, I am going to create five report, which were defining by me at question 1.

Report 1 -: get total confirmed, deaths, recovered for given country.

```
SELECT
    c.Country AS "Country",
    c.totalInfected() AS "Total Infected",
    c.totalDead() AS "Total Deaths",
    c.totalRecovered() AS "Total Recovered"
FROM countries c
WHERE c.Country = 'Sri Lanka'
GROUP BY c.Country, c.totalInfected(), c.totalDead(), c.totalRecovered()
```

```
SQL> SELECT
  2          c.Country AS "Country",
  3          c.totalInfected() AS "Total Infected",
  4          c.totalDead() AS "Total Deaths",
  5          c.totalRecovered() AS "Total Recovered"
  6  FROM countries c
  7  WHERE c.Country = 'Sri Lanka'
  8  GROUP BY c.Country, c.totalInfected(), c.totalDead(), c.totalRecovered()
  9  /

Country                        Total Infected Total Deaths Total Recovered
------------------------------ -------------- ------------ ---------------
Sri Lanka                                  77            0               3

SQL>
```

<u>Report 2</u> -: get total number of confirmed, deaths, recovered by province for given country.

```
SELECT
    c.Province AS "Province",
    c.totalInfected() AS "Total Infected",
    c.totalDead() AS "Total Deaths",
    c.totalRecovered() AS "Total Recovered"
FROM countries c
WHERE c.Country= 'Canada'
GROUP BY c.Province, c.totalInfected(), c.totalDead(), c.totalRecovered()
```

```
SQL> SELECT
  2          c.Province AS "Province",
  3          c.totalInfected() AS "Total Infected",
  4          c.totalDead() AS "Total Deaths",
  5          c.totalRecovered() AS "Total Recovered"
  6  FROM countries c
  7  WHERE c.Country= 'Canada'
  8  GROUP BY c.Province, c.totalInfected(), c.totalDead(), c.totalRecovered()
  9  /

Province                          Total Infected Total Deaths Total Recovered
--------------------------------- -------------- ------------ ---------------
New Brunswick                                 17            0               0
Grand Princess                                10            0               0
Saskatchewan                                  26            0               0
Ontario                                      377            3               6
British Columbia                             424           10               4
Manitoba                                      18            0               0
Newfoundland and Labrador                      6            0               0
Nova Scotia                                   21            0               0
Alberta                                      195            1               0
Northwest Territories                          1            0               0
Prince Edward Island                           2            0               0

Province                          Total Infected Total Deaths Total Recovered
--------------------------------- -------------- ------------ ---------------
Quebec                                       181            5               0

12 rows selected.
```

<u>Report 3</u> -: get total number of confirmed, deaths, recovered for all countries.

```
SELECT
    c.Country AS "Country",
    c.totalInfected() AS "Total Infected",
    c.totalDead() AS "Total Deaths",
    c.totalRecovered() AS "Total Recovered"
FROM countries c
GROUP BY c.Country, c.totalInfected(), c.totalDead(), c.totalRecovered()
```

```
SQL> SELECT
  2          c.Country AS "Country",
  3          c.totalInfected() AS "Total Infected",
  4          c.totalDead() AS "Total Deaths",
  5          c.totalRecovered() AS "Total Recovered"
  6  FROM countries c
  7  GROUP BY c.Country, c.totalInfected(), c.totalDead(), c.totalRecovered()
  8  /
```

There are 384 rows created there for I have added sample answer

```
Country                        Total Infected Total Deaths Total Recovered
------------------------------ -------------- ------------ ---------------
Vietnam                                    94            0              17
Singapore                                 432            2             140
Seychelles                                  7            0               0
Belgium                                  2815           67             263
China                                       1            0               1
El Salvador                                 3            0               0
Canada                                     21            0               0
Antigua and Barbuda                         1            0               0
Australia                                 436            6               4
Serbia                                    171            1               1
US                                        114            5               0

Country                        Total Infected Total Deaths Total Recovered
------------------------------ -------------- ------------ ---------------
Guinea                                      2            0               0
```

<u>Report 4</u> -: get death ratio and recovered ratio by province for given country.

```
SELECT
    c.Country AS "Country",
    c.Province AS "Province",
    (c.totalDead()/c.totalInfected()) AS "Death Ratio",
    (c.totalRecovered()/c.totalInfected()) AS "Recovered Ratio"
FROM countries c
WHERE c.Country = 'Australia'
GROUP BY c.Country, c.Province, (c.totalDead()/c.totalInfected()),
(c.totalRecovered()/c.totalInfected())
/
```

```
SQL> SELECT
  2          c.Country AS "Country",
  3          c.Province AS "Province",
  4          (c.totalDead()/c.totalInfected()) AS "Death Ratio",
  5          (c.totalRecovered()/c.totalInfected()) AS "Recovered Ratio"
  6  FROM countries c
  7  WHERE c.Country = 'Australia'
  8  GROUP BY c.Country, c.Province, (c.totalDead()/c.totalInfected()), (c.totalRecovered()/c.totalInfected())
  9  /

Country                     Province                         Death Ratio Recovered Ratio
--------------------------- -------------------------------- ----------- --------------
Australia                   Australian Capital Territory               0              0
Australia                   Western Australia                .011111111              0
Australia                   Queensland                                 0      .036199095
Australia                   Tasmania                                   0          .1875
Australia                   New South Wales                  .013761468      .009174312
Australia                   From Diamond Princess                      0              0
Australia                   Northern Territory                         0              0
Australia                   South Australia                            0      .044776119
Australia                   Victoria                                   0      .034934498

9 rows selected.
```

<u>Report 5</u> -: get remaining patients count and percentage in a hospital and percentage of dead and recovered patient's by province for given country.

```
SELECT
    c.Country AS "Country",
    c.Province AS "Province",
    c.totalInfected() AS "Total Patients",
    (c.totalInfected() - (c.totalDead() + c.totalRecovered())) AS "Remaining patients",
    (((c.totalInfected() - (c.totalDead() + c.totalRecovered())) / c.totalInfected()) * 100)  AS "Remaining Percentage",
    ((c.totalDead()/c.totalInfected()) * 100) AS "Death Percentage",
    ((c.totalRecovered()/c.totalInfected()) * 100) AS "Recovered Percentage"
FROM countries c
WHERE c.Country = 'Canada'
GROUP BY c.Country, c.Province, c.totalInfected(), (c.totalInfected() - (c.totalDead() + c.totalRecovered())),
    (((c.totalInfected() - (c.totalDead() + c.totalRecovered())) / c.totalInfected()) * 100),
    ((c.totalDead()/c.totalInfected()) * 100), ((c.totalRecovered()/c.totalInfected()) * 100)
```

```
SQL> SELECT
  2         c.Country AS "Country",
  3         c.Province AS "Province",
  4         c.totalInfected() AS "Total Patients",
  5         (c.totalInfected() - (c.totalDead() + c.totalRecovered())) AS "Remaining patients",
  6         (((c.totalInfected() - (c.totalDead() + c.totalRecovered())) / c.totalInfected()) * 100)  AS "Remaining Percentage",
  7         ((c.totalDead()/c.totalInfected()) * 100) AS "Death Percentage",
  8         ((c.totalRecovered()/c.totalInfected()) * 100) AS "Recovered Percentage"
  9  FROM countries c
 10  WHERE c.Country = 'Canada'
 11  GROUP BY c.Country, c.Province, c.totalInfected(), (c.totalInfected() - (c.totalDead() + c.totalRecovered())),
 12           (((c.totalInfected() - (c.totalDead() + c.totalRecovered())) / c.totalInfected()) * 100),
 13           ((c.totalDead()/c.totalInfected()) * 100), ((c.totalRecovered()/c.totalInfected()) * 100)
 14  /

Country                        Province                         Total Patients Remaining patients Remaining Percentage Death Percentage Recovered Percentage
------------------------------ -------------------------------- -------------- ----------------- -------------------- ---------------- --------------------
Canada                         Grand Princess                               10                10                  100                0                    0
Canada                         Northwest Territories                         1                 1                  100                0                    0
Canada                         Saskatchewan                                 26                26                  100                0                    0
Canada                         Newfoundland and Labrador                     6                 6                  100                0                    0
Canada                         New Brunswick                                17                17                  100                0                    0
Canada                         Alberta                                     195               194           99.4871795        .512820513                    0
Canada                         Ontario                                     377               368           97.6127321        .795755968            1.59151194
Canada                         Prince Edward Island                          2                 2                  100                0                    0
Canada                         Nova Scotia                                  21                21                  100                0                    0
Canada                         British Columbia                            424               410           96.6981132        2.35849057            .943396226
Canada                         Quebec                                      181               176           97.2375691        2.76243094                    0

Country                        Province                         Total Patients Remaining patients Remaining Percentage Death Percentage Recovered Percentage
------------------------------ -------------------------------- -------------- ----------------- -------------------- ---------------- --------------------
Canada                         Manitoba                                     18                18                  100                0                    0

12 rows selected.
```

Question 8

Answers belongs to question 8, were given to all questions 1 to 7.