



**Database Systems and Data Driven Applications
(IT3031)
3rd Year, 1st Semester**

Assignment 1

Submitted to
Sri Lanka Institute of Information Technology

Bachelor of Science Special Honors Degree in Data Science

**IT17167710
D.M.J. Prathapa**

1)

Report – 1

Coronavirus COVID-19 Summary of Total Cases (worldwide Overall)

This report contains about the total number of cases all around the world.

Covid-19 Total Cases = total cumulative count (304,524). This figure therefore includes recovered, deaths and currently infected patients.

According to my analysis

12,973 people have died, 91,499 recovered, and 200,052 people are currently infected so far from the coronavirus COVID-19 outbreak as of 22-MAR- 2020

Report -2

Coronavirus COVID-19 Mortality Rate /case fatality rate Report - (country and state wise)

This report contains countries and states and the number of death cases and the rate of Mortality which means this report represents number of death cases and the percentage (%) of death cases who eventually die from a **COVID-19** disease.)

The Mortality rate, which is known as **case fatality rate** can be calculated by dividing current total deaths from current confirmed cases, we can get the proportion of Mortality and multiply by hundred we can get the **rate (%) of Mortality**.

Report -3

Coronavirus COVID-19 Recovered / Discharged- (country and state wise)

This report contains countries and states and the number of recovery cases (Discharged patients) and the **rate of recovery**.

by dividing **recovery** cases from the total of confirmed cases, we can get the **proportion of recovery cases** and multiply by hundred we can get the **rate (%) of recovery cases**

Report -4

Coronavirus COVID-19 CLOSED Cases - (country and state wise)

this report mainly considers about COVID-19 Cases which had an outcome, which means total of Recovered (Discharged) and death cases.

By adding recoveries and deaths count, we get " **CLOSED Cases count**. And adding both **rate of recovery** and **Mortality Rate** together or by dividing **CLOSED Cases count** from the total of confirmed cases. we get the **rate (%) of CLOSED Cases**

Report -5

Coronavirus COVID-19 Active Cases - (country and state wise)

This report represents countries, states and the number of Currently Infected Patients and the percentage (%) of Currently Infected Patients which are affected by **COVID-19** disease.)

By reducing recoveries and deaths count from confirmed total cases, we get "currently infected cases" or "active cases". And by dividing active cases from the total of confirmed cases, we can get the **proportion of active cases** and multiply by hundred we can get the **rate (%) of active cases**

Report-6

Coronavirus COVID-19 Cases - (Coronavirus Cases in China vs Outside of China)

A. (Coronavirus Cases in China)

This report represents the total number of deaths, Active cases, recovery cases and Confirmed cases of covid 19 in china on the date of 21th March 2020.

B. (Coronavirus Cases outside of China)

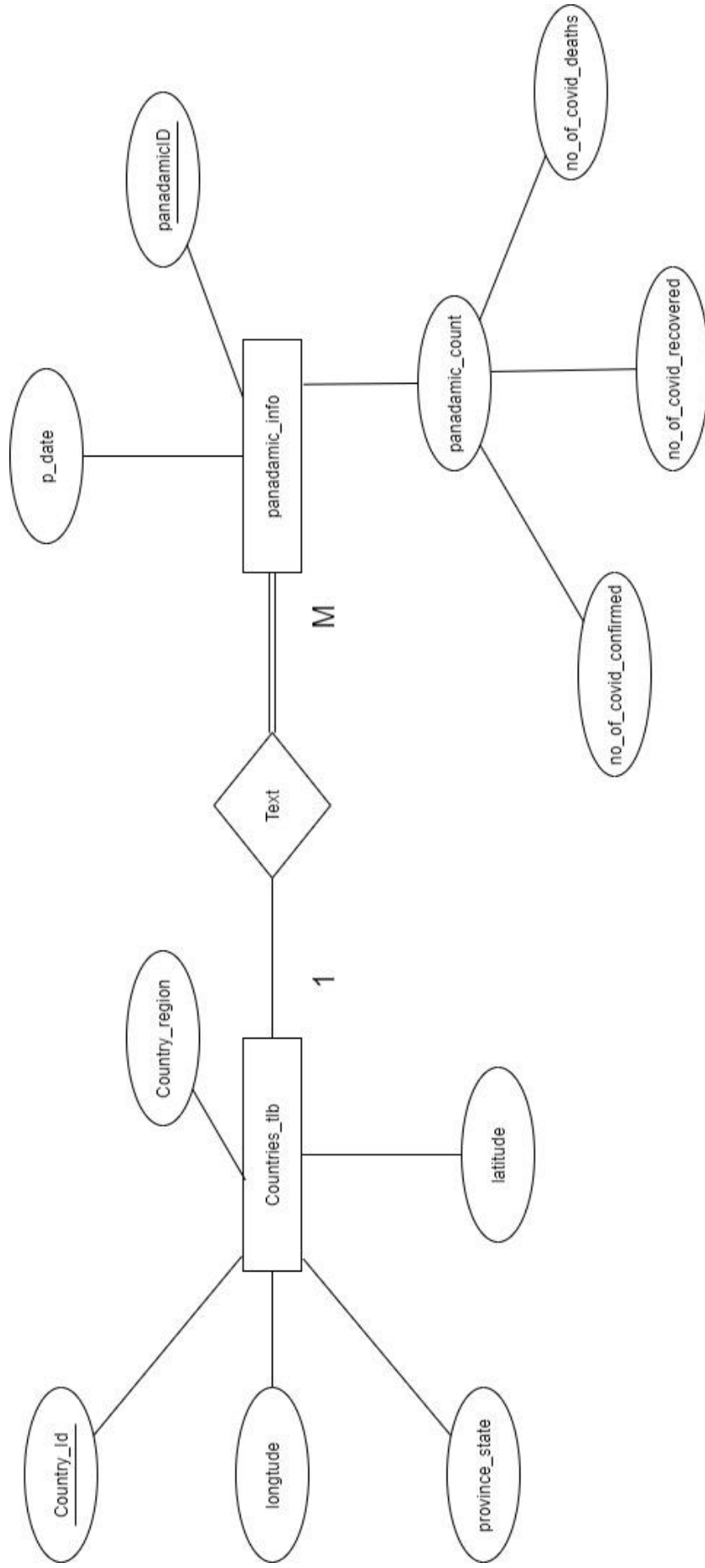
This report represents Total Coronavirus confirmed Cases, Active Cases, Deaths cases, recovery cases and closed cases in outside of China

Report-7

Coronavirus COVID-19 Cases – Comparisons

This report contains about full detailed report of the countries which exceeded 10000 patients of **COVID-19**. And each state of them where death toll is reported more than the recovery toll on 21th March,2020

2)

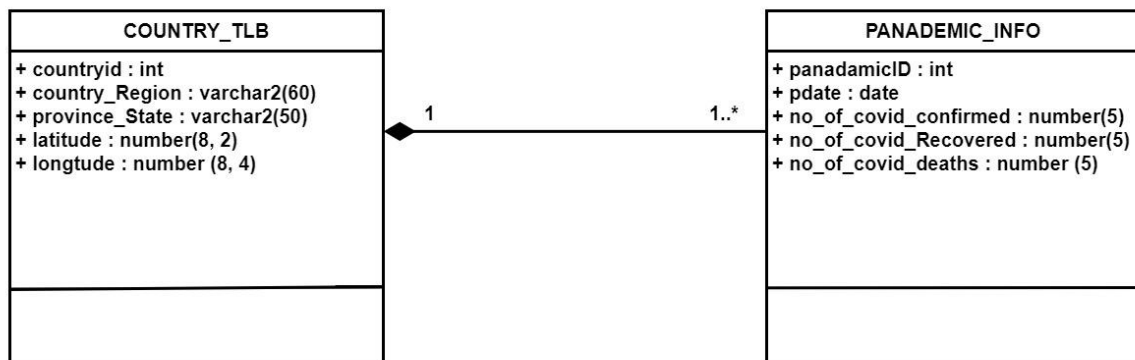


3)

Object relation model

```
countries_t(countryid:int, country_Region:vchar2(60), province_State:vchar2(50),  
            latitude:number(8,4), longitude:number(8,4) )  
  
panademic_info_t (panademicID:int, cid:ref countries_t, pdate:date, no_of_covid_confirmed:number(5),  
                  no_of_covid_Recovered:number(5), no_of_covid_deaths:number(5))
```

Object diagram



4)

Types

- To create and describe the table **countries_tlb** that contains the columns countryid, country_Region, province_State, latitude and longitude, enter.

```
CREATE TYPE countries_t AS OBJECT (  
  countryid int,  
  country_Region varchar2(60),  
  province_State varchar2(50),  
  latitude number(8,4),  
  longitude number(8,4)  
)  
/
```

- To create and describe the table **panademic_info** that contains the columns cid, pdate, province_State, no_of_covid_confirmed, no_of_covid_Recovered and no_of_covid_deaths.

```
create TYPE panademic_info_t as object(  
  panademicID int,  
  cid ref countries_t,  
  pdate date,  
  no_of_covid_confirmed number(5),  
  no_of_covid_Recovered number(5),  
  no_of_covid_deaths number(5)  
)  
/
```

-tables-

```
create table countries_tlb of countries_t(  
  countryid primary key  
)  
/  
  
create table panademic_info of panademic_info_t(  
  panademicID primary key  
)  
/
```

5)

- Create a folder called **dsda_it17167710_mid** and add given 3 CSV files into that.
- Then I grant the read write permission to the user by executing following commands as SYSTEM user. 'c:\dsda_it17167710_mid\' must be a physical path on the disk.

```
conn / as sysdba
```

```
CREATE OR REPLACE DIRECTORY MY_DIRECTORY AS 'c:\dsda_it17167710_mid\';
```

```
GRANT READ, WRITE ON DIRECTORY MY_DIRECTORY TO janith_dsds;
```

```
GRANT EXECUTE ON SYS.utl_file TO janith_dsds;
```

```
SQL>
SQL> CREATE OR REPLACE DIRECTORY MY_DIRECTORY AS 'c:\dsda_mid_it17167710\';
Directory created.
SQL> GRANT READ, WRITE ON DIRECTORY MY_DIRECTORY TO it17167710;
Grant succeeded.
SQL> GRANT EXECUTE ON SYS.UTL_FILE TO jani;
Grant succeeded.
```

- Then I renamed CSV files as covid_Confirmed, covid_Deaths, covid_Recovered and replace all empty fields in the Province/state column by using value NULL.

The screenshot shows a Microsoft Excel spreadsheet titled "time_series_19-covid-Confirmed - Excel". The spreadsheet has columns A through O. Column A is labeled "Province/ Country/R" and contains various country names. Column B is labeled "Lat" and contains latitude values. Column C is labeled "Long" and contains longitude values. Columns D through O contain dates from 1/22/20 to 1/31/20, with the last column labeled "#####". The data shows the number of confirmed COVID-19 cases for each country over time. A "Find and Replace" dialog box is open, showing the "Find" tab. The "Find what:" field is empty, and the "Replace with:" field is empty. The "Find" button is highlighted. A message box from Microsoft Excel is also visible, stating "All done. We made 157 replacements." with an "OK" button.

Province/ Country/R	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	#####
NULL Thailand	15	101	2	3									
NULL Japan	36	138	2	1									
NULL Singapore	1.2833	103.8333	0	1									
NULL Nepal	28.1667	84.25	0	0									
NULL Malaysia	2.5	112.5	0	0									
British Co Canada	49.2827	-123.121	0	0									
New Sout Australia	-33.8688	151.2093	0	0									
Victoria Australia	-37.8136	144.9631	0	0									
Queensla Australia	-28.0167	153.4	0	0									
NULL Cambodia	11.55	104.9167	0	0									
NULL Sri Lanka	7	81	0	0		0	0	1	1	1	1	1	
NULL Germany	51	9	0	0	0	0	0	1	4	4	4	5	
NULL Finland	64	26	0	0	0	0	0	0	0	1	1	1	
NULL United Ar	24	54	0	0	0	0	0	0	0	4	4	4	

- To Load " covid_Confirmed.CSV, covid_Deaths.CSV, covid_Recovered.CSV " file into Oracle Tables I have created all the tables and

```
SQL>
SQL> CREATE TYPE countries_t AS OBJECT (
  2  countryid int,
  3  country_Region varchar2(60),
  4  province_State varchar2(50),
  5  latitude number(8,4),
  6  longtude number(8,4)
  7  )
  8  /

Type created.

SQL>
SQL> create TYPE panademic_info_t as object(
  2  panademicID int,
  3  cid ref countries_t,
  4  pdate date,
  5  no_of_covid_confirmed number(5),
  6  no_of_covid_Recovered number(5),
  7  no_of_covid_deaths number(5)
  8  )
  9  /

Type created.

SQL> create table countries_tlb of countries_t(
  2  countryid primary key
  3  )
  4  /

Table created.

SQL>
SQL> create table panademic_info of panademic_info_t(
  2  panademicID primary key
  3  )
  4  /

Table created.
```


- Once the tables and types are created, Then I executed the following pl/sql code block to get the all dates into a separate CSV file called **covid_dates.CSV**.

```

DECLARE
    F UTL_FILE.FILE_TYPE;
    WRITE_FILE UTL_FILE.FILE_TYPE;
    covid_LINE VARCHAR2 (1000);
    covid_date VARCHAR2 (50);

    i number(5);
BEGIN
    F :=
    UTL_FILE.FOPEN('MY_DIRECTORY','c:\dsda_mid_it17167710\covid_Confirmed.CSV','R');
    WRITE_FILE
    :=UTL_FILE.FOPEN('MY_DIRECTORY','c:\dsda_mid_it17167710\covid_dates.CSV','W');

    IF UTL_FILE.IS_OPEN(F) THEN
        LOOP
            BEGIN

                UTL_FILE.GET_LINE(F, covid_LINE, 1000);
                IF covid_LINE IS NULL THEN
                    EXIT;
                END IF;
                i :=5;
                LOOP
                    BEGIN

                        covid_date := replace(REGEXP_SUBSTR(covid_LINE,
                        '("([^\"]*)")|[\^,]+' , 1, i),'', '');
                        UTL_FILE.PUT_LINE(WRITE_FILE,covid_date);

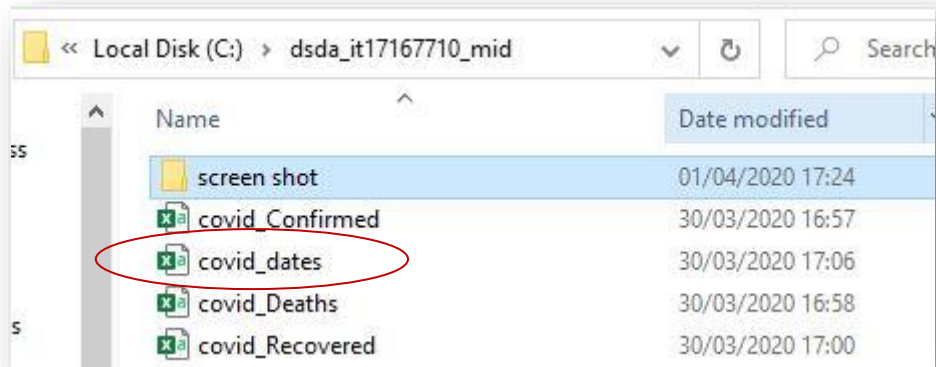
                        COMMIT;
                        i:=i+1;
                        IF covid_date IS NULL THEN
                            EXIT;
                        END IF;
                    END;
                END LOOP;
                EXIT;
            END LOOP;
        END IF;
    END IF;

```

Read the
covid_Confirmed.CSV
file

Write dates in to
covid_dates.CSV file

- After the execution of above procedure, a file (**covid_dates.CSV**) would have been created at '**c:\dsda_it17167710_mid**'location.



- to create an incrementing numeric key, I create a sequence.

```
create SEQUENCE seq1  
start with 1  
increment by 1;
```

- then I define a trigger called **covid_countries_trigger** that automatically populates the primary key value using the sequence **seq1**

```
CREATE OR REPLACE TRIGGER covid_countries_trigger  
BEFORE INSERT  
ON countries_tlb  
REFERENCING NEW AS NEW  
FOR EACH ROW  
BEGIN  
SELECT seq1.nextval INTO :NEW.countryid FROM dual;  
END;  
/
```

Then I do the same thing for the `panademic_info` table.

```
create SEQUENCE seq2
start with 1
increment by 1;
```

- then I executed the bellow a trigger called `covid_panademic_trigger` that automatically populates the primary key value using the sequence `seq2`

```
CREATE OR REPLACE TRIGGER covid_panademic_trigger
BEFORE INSERT
ON panademic_info
REFERENCING NEW AS NEW
FOR EACH ROW
BEGIN
SELECT seq1.nextval INTO :NEW.panademicID FROM dual;
END;
/
```

- Then I change the date format by using bellow session because in the previous attempt I got an error called `ORA-01843: "not a valid month"`.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'MM/DD/YYYY';
```

After that I create one more procedure to, load CSV data to the both tables. By executing the following pl/sql block date load the CSV file into the Oracle table.

DECLARE

```
ConfirmedCSV UTL_FILE.FILE_TYPE;  
DeathsCSV UTL_FILE.FILE_TYPE;  
RecoveredCSV UTL_FILE.FILE_TYPE;  
WIRTE_FILE UTL_FILE.FILE_TYPE;  
covid_LINE_C VARCHAR2 (2000);  
covid_LINE_D VARCHAR2 (2000);  
covid_LINE_R VARCHAR2 (2000);  
covid_LINE2 VARCHAR2 (2000);  
covid_date VARCHAR2 (50);  
covid_Confirmed NUMBER(7);  
covid_Deaths NUMBER(7);  
covid_Recoverd NUMBER(7);  
covid_state VARCHAR2 (50);  
covid_region VARCHAR2 (50);  
covid_lat NUMBER(8,4);  
covid_long NUMBER(8,4);
```

```
    i number(5);  
    row number(5);  
BEGIN
```

```
ConfirmedCSV := UTL_FILE.FOPEN  
('MY_DIRECTORY','c:\dsda_mid_it17167710\covid_Confirmed.CSV', 'R');
```

```
DeathsCSV := UTL_FILE.FOPEN  
('MY_DIRECTORY','c:\dsda_mid_it17167710\covid_Deaths.CSV', 'R');
```

```
RecoveredCSV := UTL_FILE.FOPEN  
('MY_DIRECTORY','c:\dsda_mid_it17167710\covid_Recovered.CSV', 'R');
```

```
    i:=0;  
    row:=0;
```

```
    IF UTL_FILE.IS_OPEN(ConfirmedCSV) AND UTL_FILE.IS_OPEN(DeathsCSV) AND  
    UTL_FILE.IS_OPEN(RecoveredCSV) THEN
```

```
        LOOP
```

```
            BEGIN
```

```
                i:=i+1;
```

```
                UTL_FILE.GET_LINE(ConfirmedCSV, covid_LINE_C, 2000);
```

```
                UTL_FILE.GET_LINE(DeathsCSV, covid_LINE_D, 2000);
```

```
                UTL_FILE.GET_LINE(RecoveredCSV, covid_LINE_R, 2000);
```

```
                IF i =1 THEN
```

```
                    CONTINUE;
```

```
                END IF;
```

```
                IF covid_LINE_C IS NULL THEN
```

```
                    EXIT;
```

```
                END IF;
```

```

i :=5;

covid_state := replace(REGEXP_SUBSTR(covid_LINE_C,
'("([^\"]*)")|([^\,]+', 1, 1), ''', ''));
covid_region := REGEXP_SUBSTR(covid_LINE_C, '("([^\"]*)")|([^\,]+',
1, 2);
covid_lat := REGEXP_SUBSTR(covid_LINE_C, '("([^\"]*)")|([^\,]+', 1,
3);
covid_long := REGEXP_SUBSTR(covid_LINE_C, '("([^\"]*)")|([^\,]+', 1,
4);

INSERT INTO countries_tlb
VALUES(0,covid_region,covid_state,covid_lat,covid_long);
COMMIT;
row :=row+1;
WIRTE_FILE := UTL_FILE.FOPEN
('MY_DIRECTORY','c:\dsda_mid_it17167710\covid_dates.CSV', 'R',2000);
IF UTL_FILE.IS_OPEN(WIRTE_FILE) THEN

LOOP
BEGIN
UTL_FILE.GET_LINE(WIRTE_FILE, covid_LINE2, 100);
IF covid_LINE2 IS NULL THEN
UTL_FILE.FCLOSE(WIRTE_FILE);
EXIT;
END IF;
covid_date := REGEXP_SUBSTR(covid_LINE2,
'("([^\"]*)")|([^\,]+', 1, 1);
covid_Confirmed := REGEXP_SUBSTR(covid_LINE_C,
'("([^\"]*)")|([^\,]+', 1, i);
covid_Deaths := REGEXP_SUBSTR(covid_LINE_D,
'("([^\"]*)")|([^\,]+', 1, i);
covid_Recoverd := REGEXP_SUBSTR(covid_LINE_R,
'("([^\"]*)")|([^\,]+', 1, i);

INSERT INTO panademic_info values(0,(select ref(c) from countries_tlb c
where c.countryid=row) , covid_date ,
covid_Confirmed,covid_Recoverd,covid_Deaths);
COMMIT;
i:=i+1;

END;
END LOOP;

END IF;

EXCEPTION
WHEN NO_DATA_FOUND THEN
EXIT;
END;
END LOOP;
END IF;
IF UTL_FILE.is_open(WIRTE_FILE) THEN
UTL_FILE.fclose(WIRTE_FILE);
END IF;

UTL_FILE.FCLOSE(ConfirmedCSV);
UTL_FILE.FCLOSE(DeathsCSV);
UTL_FILE.FCLOSE(RecoveredCSV);
END;
/

```

insert query for load data to
countries_tlb

insert query for load data to
panademic_info

Data of "countries_tlb" and "panademic_info" can be determined as following

Worksheet Query Builder

```
select *
from countries_tlb;
```

Script Output x Query Result x

SQL | All Rows Fetched: 482 in 0.059 seconds

	COUNTRYID	COUNTRY_REGION	PROVINCE_STATE	LATITUDE	LONGITUDE
464	262	US	Fresno County, CA	36.9859	-119.2321
465	263	US	Harford County, MD	39.5839	-76.3637
466	264	US	Hendricks County, IN	39.8065	-86.5401
467	265	US	Hudson County, NJ	40.7453	-74.0535
468	266	US	Johnson County, KS	38.8454	-94.8521
469	267	US	Kittitas County, WA	47.175	-120.9319
470	268	US	Manatee County, FL	27.4799	-82.3452
471	269	US	Marion County, OR	44.8446	-122.5927
472	270	US	Okaloosa County, FL	30.5773	-86.6611
473	271	US	Polk County, GA	34.0132	-85.1479
474	272	US	Riverside County, CA	33.9533	-117.3961
475	273	US	Shelby County, TN	35.1269	-89.9253
476	274	US	St. Louis County, MO	38.6103	-90.4125
477	275	US	Suffolk County, NY	40.9849	-72.6151
478	276	US	Ulster County, NY	41.8586	-74.3118
479	277	US	Volusia County, FL	29.028	-81.0755
480	278	US	Fairfax County, VA	38.9085	-77.2405
481	279	US	Rockingham County, NH	42.9931	-71.0498
482	280	US	Washington, D.C.	38.9072	-77.0369

Query Builder

```
select *FROM panademic_info;
```

Result x

SQL | Fetched 850 rows in 0.911 seconds

PANADAMICID	CID	PDATE	NO_OF_COVID_CONFI...	NO_OF_COVID_RECOVERED	NO_OF_COVID_DEATHS
1	[JANITH1.COUNTRIES_I]	22-JAN-20	2	0	0
2	[JANITH1.COUNTRIES_I]	23-JAN-20	3	0	0
3	[JANITH1.COUNTRIES_I]	24-JAN-20	5	0	0
4	[JANITH1.COUNTRIES_I]	25-JAN-20	7	0	0
5	[JANITH1.COUNTRIES_I]	26-JAN-20	8	2	0
6	[JANITH1.COUNTRIES_I]	27-JAN-20	8	2	0
7	[JANITH1.COUNTRIES_I]	28-JAN-20	14	5	0
8	[JANITH1.COUNTRIES_I]	29-JAN-20	14	5	0
9	[JANITH1.COUNTRIES_I]	30-JAN-20	14	5	0
10	[JANITH1.COUNTRIES_I]	31-JAN-20	19	5	0
11	[JANITH1.COUNTRIES_I]	01-FEB-20	19	5	0
12	[JANITH1.COUNTRIES_I]	02-FEB-20	19	5	0
13	[JANITH1.COUNTRIES_I]	03-FEB-20	19	5	0
14	[JANITH1.COUNTRIES_I]	04-FEB-20	25	5	0
15	[JANITH1.COUNTRIES_I]	05-FEB-20	25	5	0
16	[JANITH1.COUNTRIES_I]	06-FEB-20	25	5	0
17	[JANITH1.COUNTRIES_I]	07-FEB-20	25	5	0
18	[JANITH1.COUNTRIES_I]	08-FEB-20	32	10	0
19	[JANITH1.COUNTRIES_I]	09-FEB-20	32	10	0
20	[JANITH1.COUNTRIES_I]	10-FEB-20	32	10	0

6) Member methods

- **MEMBER FUNCTION 1:** calculate the number of Closed Cases

number of Closed Cases = `Recovered count` + `deaths count`

- **MEMBER FUNCTION 2:** calculate the number of Active cases

number of Active cases = `Confirmed patients- Recovered patients- Died patients`

- **MEMBER FUNCTION 3:** calculate the Mortality Rate

Mortality Rate (The case fatality rate) = (`deaths count` / `Confirmed count`) * 100

- **MEMBER FUNCTION 4:** calculate the Active cases Rate

Active cases Rate = (`number of Active Cases` / `Confirmed count`) * 100

- **MEMBER FUNCTION 5:** calculate the Recovery cases Rate

Recovery cases Rate = (`number of Recovered Cases` / `Confirmed count`) * 100

```
ALTER TYPE panademic_info_t
ADD MEMBER FUNCTION numberOf_CLOSED_CASES
RETURN NUMBER CASCADE;
```

```
ALTER TYPE panademic_info_t
ADD MEMBER FUNCTION numberOf_Active_Cases
RETURN NUMBER CASCADE;
```

```
ALTER TYPE panademic_info_t
ADD MEMBER FUNCTION covid_mortality_Rate
RETURN FLOAT CASCADE;
```

```
ALTER TYPE panademic_info_t
ADD MEMBER FUNCTION Active_Cases_Rate
RETURN FLOAT CASCADE;
```

```
ALTER TYPE panademic_info_t
ADD MEMBER FUNCTION covid_recover_Rate
RETURN FLOAT CASCADE;
```

```

CREATE OR REPLACE TYPE BODY panademic_info_t
AS MEMBER FUNCTION
numberOf_CLOSED_CASES
RETURN NUMBER IS
BEGIN
    RETURN SELF.no_of_covid_Recovered + SELF.no_of_covid_deaths;
END numberOf_CLOSED_CASES;

MEMBER FUNCTION
numberOf_Active_Cases
RETURN NUMBER IS
BEGIN
    RETURN SELF.no_of_covid_confirmed -SELF.no_of_covid_Recovered
-Self.no_of_covid_deaths ;
END numberOf_Active_Cases ;

MEMBER FUNCTION covid_mortality_Rate
RETURN FLOAT IS
    confirmed NUMBER;
BEGIN
    confirmed :=1;
    IF self.no_of_covid_confirmed >0 THEN
        confirmed :=self.no_of_covid_confirmed;
    END IF;

    RETURN (self.no_of_covid_deaths /confirmed)*100;

END covid_mortality_Rate;

MEMBER FUNCTION Active_Cases_Rate
RETURN FLOAT IS
    confirmed NUMBER;
    x NUMBER;

BEGIN
    confirmed :=1;
    IF self.no_of_covid_confirmed >0 THEN
        confirmed :=self.no_of_covid_confirmed;
    END IF;

    x:= SELF.no_of_covid_confirmed -SELF.no_of_covid_Recovered -
SELF.no_of_covid_deaths ;

    RETURN (x /confirmed)*100;

END Active_Cases_Rate;

```



```
MEMBER FUNCTION covid_recover_Rate
RETURN FLOAT IS
    confirmed NUMBER;
BEGIN
    confirmed :=1;
    IF self.no_of_covid_confirmed >0 THEN
        confirmed :=self.no_of_covid_confirmed;
    END IF;

    RETURN (self.no_of_covid_Recovered /confirmed)*100;

END covid_recover_Rate;

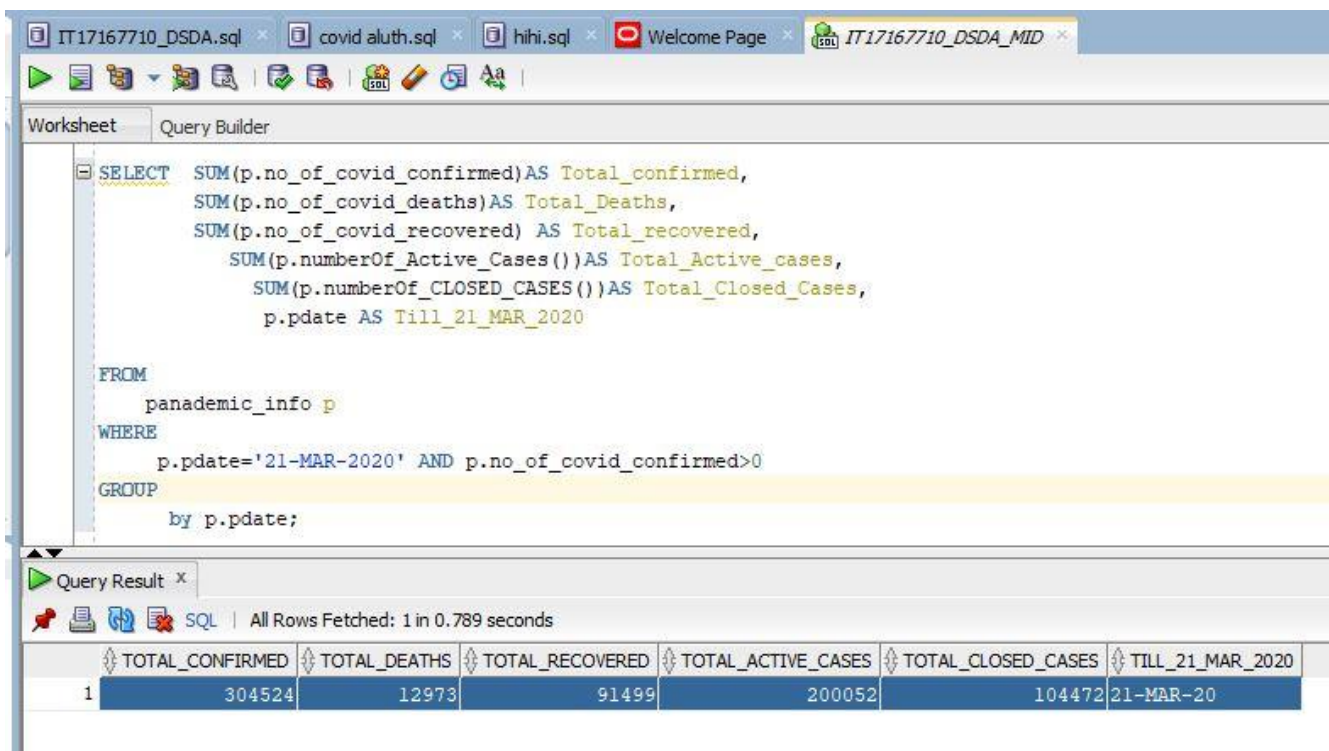
END;
```

7) OR-SQL statement to produce 5 reports

1. Coronavirus COVID-19 Summary of Total Cases (worldwide Overall)

```
SELECT SUM(p.no_of_covid_confirmed)AS Total_confirmed,
       SUM(p.no_of_covid_deaths)AS Total_deaths,
       SUM(p.no_of_covid_recovered) AS Total_recovered,
       SUM(p.numberOf_Active_Cases())AS Total_Active_cases,
       SUM(p.numberOf_CLOSED_CASES())AS Total_Closed_Cases,
       p.pdate AS Till_21_MAR_2020

FROM   panademic_info p
WHERE  p.pdate='21-MAR-2020' AND p.no_of_covid_confirmed>0
GROUP  by p.pdate;
```



Query Result x

SQL | All Rows Fetched: 1 in 0.789 seconds

	TOTAL_CONFIRMED	TOTAL_DEATHS	TOTAL_RECOVERED	TOTAL_ACTIVE_CASES	TOTAL_CLOSED_CASES	TILL_21_MAR_2020
1	304524	12973	91499	200052	104472	21-MAR-20

According to this report 12,973 people have died, 91,499 recovered, and 200,052 people are currently infected so far from the coronavirus COVID-19 outbreak as of 22-MAR- 2020

Number of Closed Cases = Recovered count + deaths count

$$\begin{array}{r} 91,499 + 12,973 \\ \hline 104,472 \\ \hline \end{array}$$

Number of Confirmed Cases = total deaths cases + total recovery cases + total active cases

$$\begin{array}{r} 12,973 + 91,499 + 200,052 \\ \hline 304,524 \\ \hline \end{array}$$

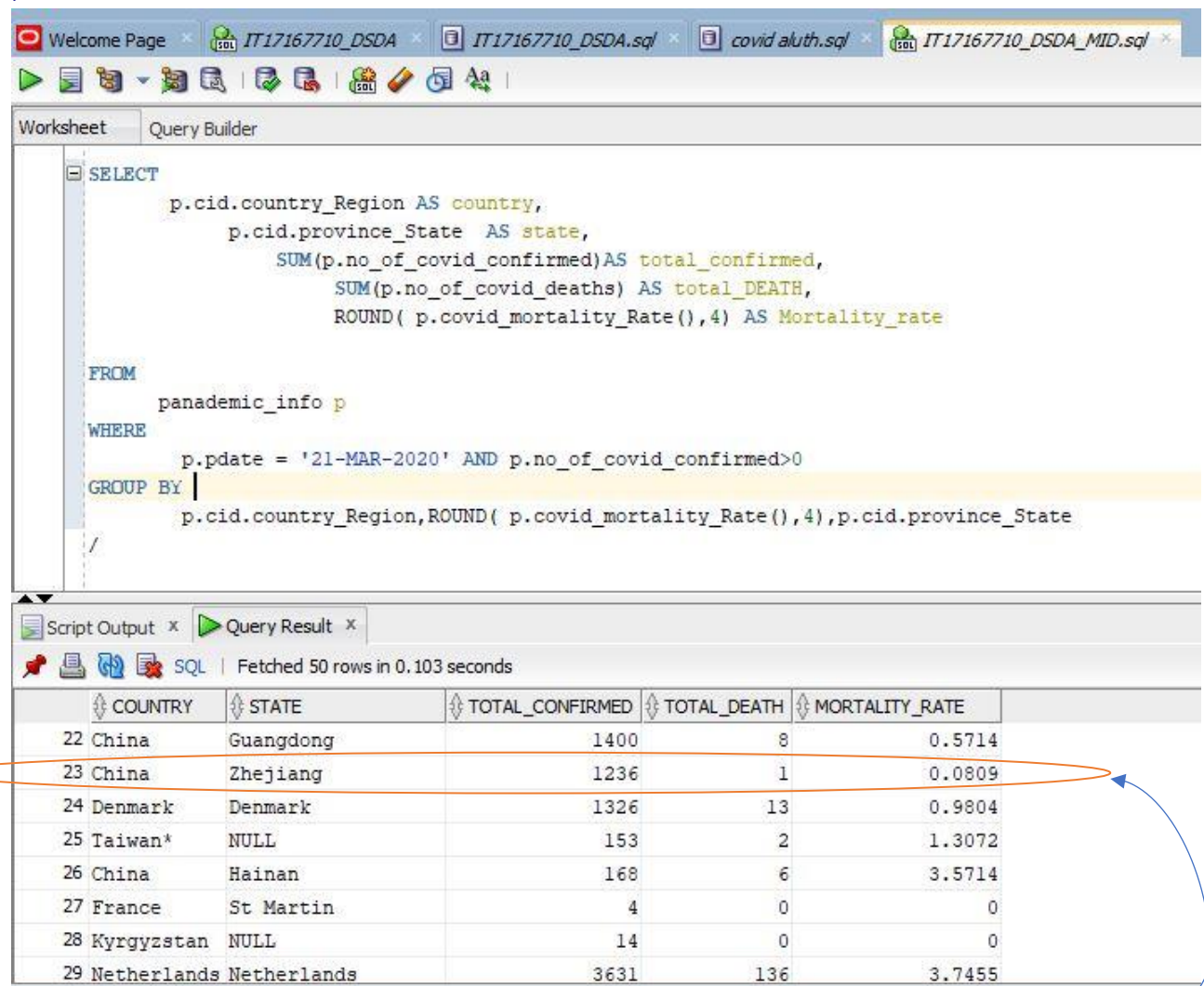
2. Coronavirus COVID-19 Mortality Rate /case fatality rate Report - (country and state wise)

```
SELECT
    p.cid.country_Region AS country,
    p.cid.province_State AS state,
    SUM(p.no_of_covid_confirmed) AS total_confirmed,
    SUM(p.no_of_covid_deaths) AS total_DEATH,
    ROUND( p.covid_mortality_Rate(),4) AS Mortality_rate

FROM
    panademic_info p

WHERE
    p.pdate = '21-MAR-2020' AND p.no_of_covid_confirmed>0

GROUP BY
    p.cid.country_Region,ROUND(
p.covid_mortality_Rate(),4),p.cid.province_State
/
```



	COUNTRY	STATE	TOTAL_CONFIRMED	TOTAL_DEATH	MORTALITY_RATE
22	China	Guangdong	1400	8	0.5714
23	China	Zhejiang	1236	1	0.0809
24	Denmark	Denmark	1326	13	0.9804
25	Taiwan*	NULL	153	2	1.3072
26	China	Hainan	168	6	3.5714
27	France	St Martin	4	0	0
28	Kyrgyzstan	NULL	14	0	0
29	Netherlands	Netherlands	3631	136	3.7455

$$\begin{aligned}
 \text{Mortality Rate (The case fatality rate)} &= (\text{deaths count} / \text{Confirmed count}) * 100 \\
 &= (1 / 1236) * 100 \\
 &= 0.0008090 * 100 \\
 &= \underline{\underline{0.08090\%}}
 \end{aligned}$$

In China Zhejiang state, about 0.08090% of reported COVID-19 cases have died

3. Coronavirus COVID-19 Recovered / Discharged- (country and state wise)

SELECT

```
p.cid.country_Region AS country,
p.cid.province_State AS state,
SUM(p.no_of_covid_confirmed)AS total_confirmed,
SUM(p.no_of_covid_recovered) AS total_recovered,
ROUND( p.covid_recover_Rate(),4) AS Recovery_rate
```

FROM

```
panademic_info p
```

WHERE

```
p.pdate = '21-MAR-2020' AND p.no_of_covid_confirmed>0
```

GROUP BY

```
p.cid.country_Region,ROUND(
p.covid_recover_Rate(),4),p.cid.province_State
/
```

	COUNTRY	STATE	TOTAL_CONFIRMED	TOTAL_RECOVERED	RECOVERY_RATE
22	China	Jiangsu	631	631	100
23	China	Guangxi	254	250	98.4252
24	Denmark	Denmark	1326	1	0.0754
25	China	Beijing	504	396	78.5714
26	China	Guizhou	146	144	98.6301
27	France	St Martin	4	0	0
28	US	Puerto Rico	21	0	0
29	Kyrgyzstan	NULL	14	0	0

$$\begin{aligned}
 \text{Recovery cases Rate} &= (\text{number of Recovered Cases} / \text{Confirmed count}) * 100 \\
 &= (396 / 504) * 100 \\
 &= 0.78571 * 100 \\
 &= \underline{78.5714 \%}
 \end{aligned}$$

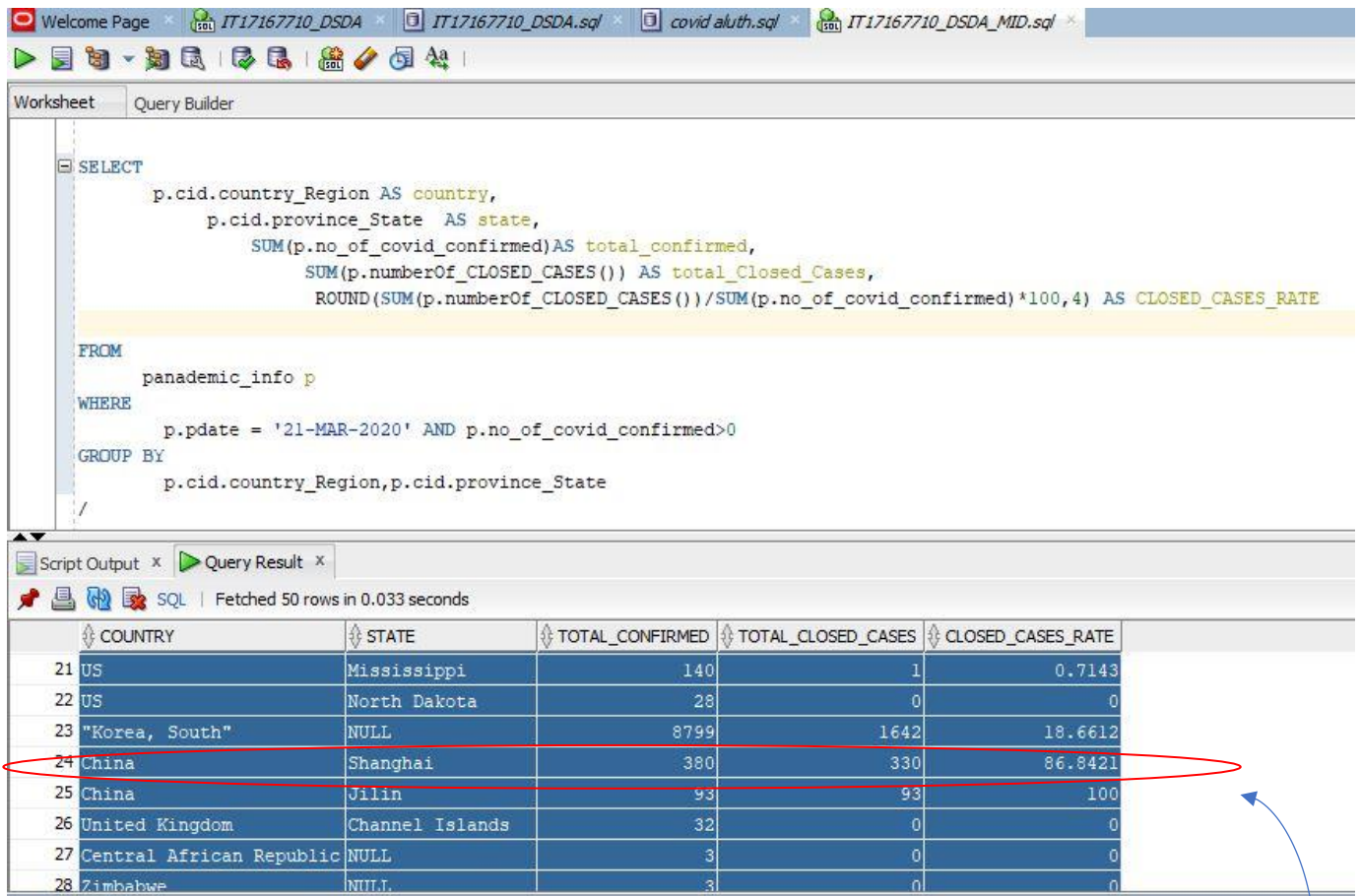
In China Beijing state, about 78.5714 % of reported COVID-19 cases have Recovered

Coronavirus COVID-19 CLOSED Cases - (country and state wise)

```
SELECT
    p.cid.country_Region AS country,
    p.cid.province_State AS state,
    SUM(p.no_of_covid_confirmed) AS total_confirmed,
    SUM(p.numberOf_CLOSED_CASES()) AS total_Closed_Cases,

    ROUND (SUM(p.numberOf_CLOSED_CASES()) / SUM(p.no_of_covid_confirmed) * 100, 4) AS
    CLOSED_CASES_RATE

FROM
    panademic_info p
WHERE
    p.pdate = '21-MAR-2020' AND p.no_of_covid_confirmed > 0
GROUP BY
    p.cid.country_Region, p.cid.province_State
/
```



	COUNTRY	STATE	TOTAL_CONFIRMED	TOTAL_CLOSED_CASES	CLOSED_CASES_RATE
21	US	Mississippi	140	1	0.7143
22	US	North Dakota	28	0	0
23	"Korea, South"	NULL	8799	1642	18.6612
24	China	Shanghai	380	330	86.8421
25	China	Jilin	93	93	100
26	United Kingdom	Channel Islands	32	0	0
27	Central African Republic	NULL	3	0	0
28	Zimbabwe	NULL	3	0	0

Closed Cases Rate = (number of Closed Cases / Confirmed count) * 100

$$= (330 / 380) * 100$$

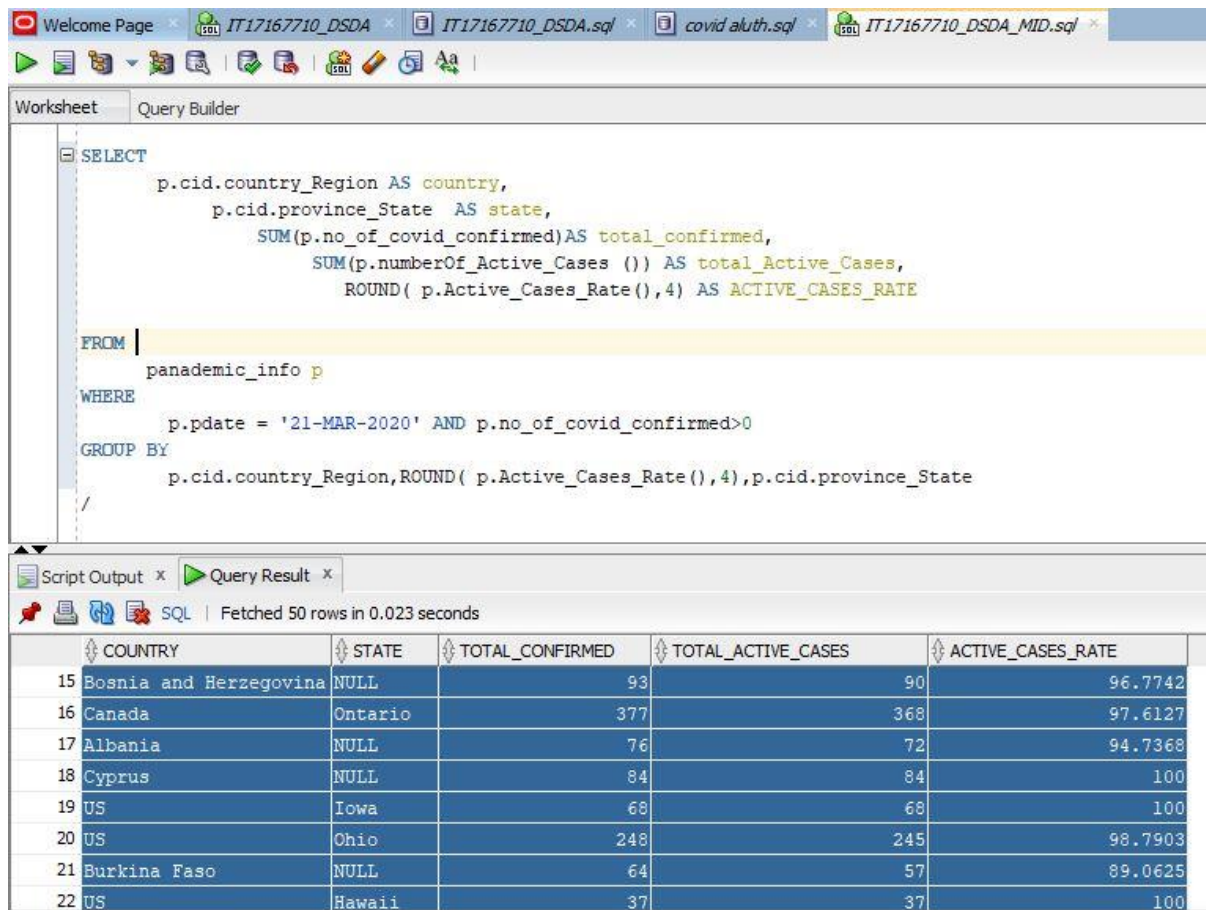
$$= 0.868421 * 100 = \underline{\underline{86.8421\%}}$$

In China Shanghai state, about **86.8421%** of reported COVID-19 cases have an outcome (which means either died or Discharged).

Coronavirus COVID-19 Active Cases - (country and state wise)

```
SELECT
    p.cid.country_Region AS country,
    p.cid.province_State AS state,
    SUM(p.no_of_covid_confirmed) AS total_confirmed,
    SUM(p.numberOf_Active_Cases ()) AS total_Active_Cases,
    ROUND( p.Active_Cases_Rate(), 4) AS
ACTIVE_CASES_RATE

FROM
    panademic_info p
WHERE
    p.pdate = '21-MAR-2020' AND p.no_of_covid_confirmed>0
GROUP BY
    p.cid.country_Region,ROUND(
p.Active_Cases_Rate(), 4),p.cid.province_State
/
```



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL code:

```
SELECT
    p.cid.country_Region AS country,
    p.cid.province_State AS state,
    SUM(p.no_of_covid_confirmed) AS total_confirmed,
    SUM(p.numberOf_Active_Cases ()) AS total_Active_Cases,
    ROUND( p.Active_Cases_Rate(), 4) AS ACTIVE_CASES_RATE

FROM
    panademic_info p
WHERE
    p.pdate = '21-MAR-2020' AND p.no_of_covid_confirmed>0
GROUP BY
    p.cid.country_Region,ROUND( p.Active_Cases_Rate(), 4),p.cid.province_State
/
```

The results window shows the following data:

	COUNTRY	STATE	TOTAL_CONFIRMED	TOTAL_ACTIVE_CASES	ACTIVE_CASES_RATE
15	Bosnia and Herzegovina	NULL	93	90	96.7742
16	Canada	Ontario	377	368	97.6127
17	Albania	NULL	76	72	94.7368
18	Cyprus	NULL	84	84	100
19	US	Iowa	68	68	100
20	US	Ohio	248	245	98.7903
21	Burkina Faso	NULL	64	57	89.0625
22	US	Hawaii	37	37	100

- In Canada Ontario, about **97.61217%** of reported COVID-19 cases are Currently Infected.
- In US, Iowa state, about **100%** of reported COVID-19 cases are Currently Infected.

Coronavirus COVID-19 Cases - (Coronavirus Cases in China vs Outside of China)

A. (Coronavirus Cases in China)

This report represents the total number of deaths, Active cases, recovery cases and Confirmed cases of covid 19 in china on the date of 21th March 2020.

```
SELECT p.pdate,
SUM(p.no_of_covid_confirmed)AS confirmed ,
sum(p.no_of_covid_deaths) AS deaths ,
ROUND(SUM(p.no_of_covid_deaths)/SUM(p.no_of_covid_confirmed)*100,4) AS
mortality_Rate,

SUM(p.no_of_covid_recovered) AS recovered,
ROUND(SUM(p.no_of_covid_recovered)/SUM(p.no_of_covid_confirmed)*100,4) AS
Recovery_Rate,
SUM(p.numberof_Active_Cases()) AS Active_cases,
ROUND((sum(p.numberof_Active_Cases())/sum(p.no_of_covid_confirmed))*100,3) AS
Active_Rate,
SUM(p. numberof_CLOSED_CASES()) AS closed_cases,
ROUND(SUM(p.numberof_CLOSED_CASES())/SUM(p.no_of_covid_confirmed)*100,4) AS
CLOSED_CASES_RATE

FROM
    panademic_info p
WHERE
    p.pdate='21-MAR-2020' AND p.no_of_covid_confirmed>0 AND p.cid.country_Region
='China'
GROUP by
    p.pdate;
```

Worksheet

Query Builder

SELECT

p.pdate,

SUM(p.no_of_covid_confirmed)AS confirmed ,

sum(p.no_of_covid_deaths) AS deaths ,

ROUND(SUM(p.no_of_covid_deaths)/SUM(p.no_of_covid_confirmed)*100,4) AS mortality_Rate,

SUM(p.no_of_covid_recovered) AS recovered,

ROUND(SUM(p.no_of_covid_recovered)/SUM(p.no_of_covid_confirmed)*100,4) AS Recovery_Rate,

SUM(p.numberof_Active_Cases()) AS Active_cases,

ROUND((sum(p.numberof_Active_Cases())/sum(p.no_of_covid_confirmed))*100,3) AS Active_Rate,

SUM(p. numberof_CLOSED_CASES()) AS closed_cases,

ROUND(SUM(p.numberof_CLOSED_CASES())/SUM(p.no_of_covid_confirmed)*100,4) AS CLOSED_CASES_RATE

FROM

panademic_info p

WHERE

p.pdate='21-MAR-2020' AND p.no_of_covid_confirmed>0 AND p.cid.country_Region = 'China'

GROUP by

p.pdate;

Query Result x

SQL | All Rows Fetched: 1 in 0.024 seconds

DPDATE	CONFIRMED	DEATHS	MOTALITY_RATE	RECOVERED	RECOVERY_RATE	ACTIVE_CASES	ACTIVE_RATE	CLOSED_CASES	CLOSED_CASES_RATE
1 21-MAR-20	81305	3259	4.0084	71857	88.3796	6189	7.612	75116	92.3879

- 81305 have confirmed, 3259 people have died with a rate of 4.0084%, 71857 recovered with a rate of 88.3796%, and 6189 people are currently infected with a rate of 7.612 %so far from the coronavirus COVID-19 outbreak as of 21-MAR- 2020 in china

C. (Coronavirus Cases outside of China)

This report represents Total Coronavirus confirmed Cases, Active Cases, Deaths cases, recovery cases and closed cases in outside of China

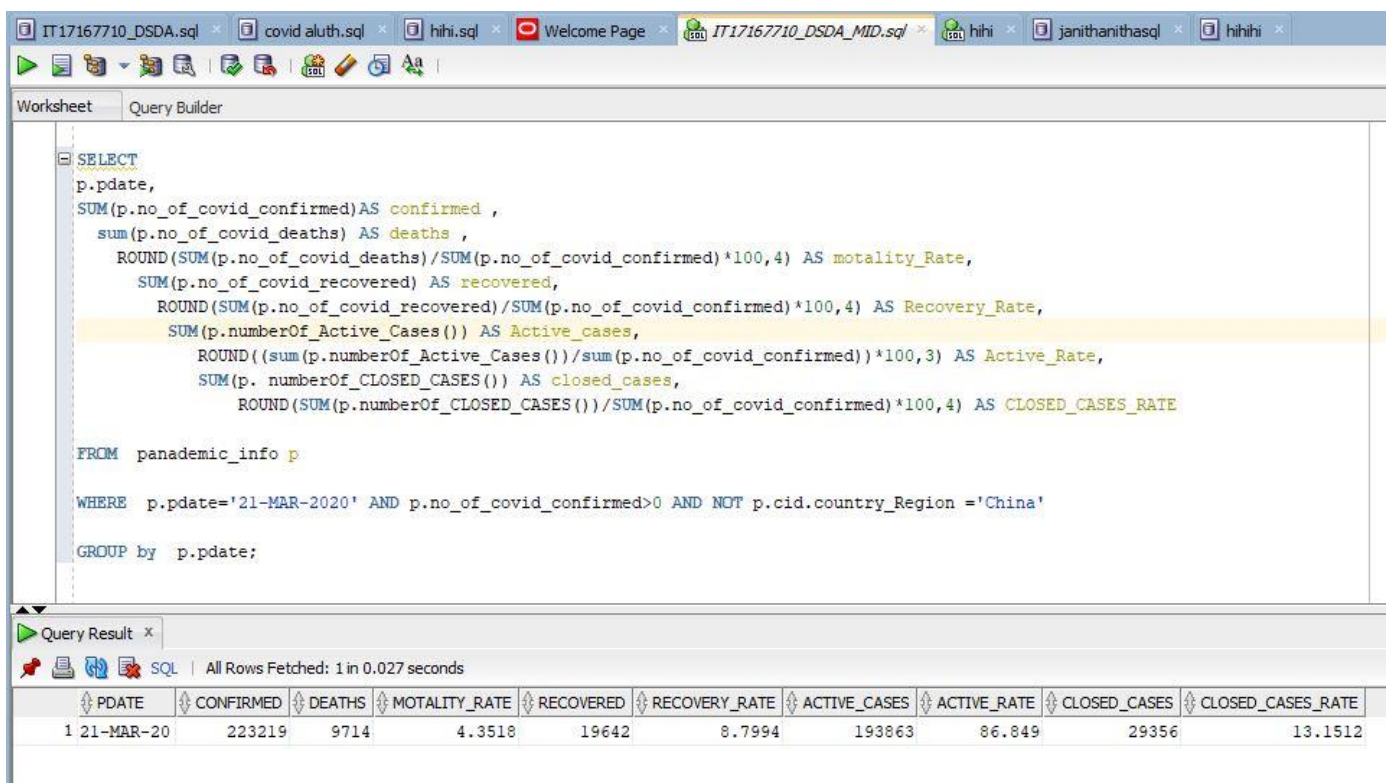
```
SELECT p.pdate,
       SUM(p.no_of_covid_confirmed) AS confirmed ,
       SUM(p.no_of_covid_deaths) AS deaths ,
       ROUND (SUM(p.no_of_covid_deaths)/SUM(p.no_of_covid_confirmed)*100,4) AS
mortality_Rate,
       SUM(p.no_of_covid_recovered) AS recovered,
       ROUND (SUM(p.no_of_covid_recovered)/SUM(p.no_of_covid_confirmed)*100,4) AS
Recovery_Rate,
       SUM(p.numberOf_Active_Cases()) AS Active_cases,
       ROUND ((sum(p.numberOf_Active_Cases())/sum(p.no_of_covid_confirmed))*100,3) AS
Active_Rate,
       SUM(p. numberOf_CLOSED_CASES()) AS closed_cases,
       ROUND (SUM(p. numberOf_CLOSED_CASES())/SUM(p.no_of_covid_confirmed)*100,4)

AS CLOSED_CASES_RATE

FROM   panademic_info p

WHERE  p.pdate='21-MAR-2020' AND p.no_of_covid_confirmed>0 AND NOT
p.cid.country_Region ='China'

GROUP by p.pdate;
```



The screenshot shows a SQL query editor with the following query:

```
SELECT
p.pdate,
SUM(p.no_of_covid_confirmed)AS confirmed ,
sum(p.no_of_covid_deaths) AS deaths ,
ROUND (SUM(p.no_of_covid_deaths)/SUM(p.no_of_covid_confirmed)*100,4) AS mortality_Rate,
SUM(p.no_of_covid_recovered) AS recovered,
ROUND (SUM(p.no_of_covid_recovered)/SUM(p.no_of_covid_confirmed)*100,4) AS Recovery_Rate,
SUM(p.numberOf_Active_Cases()) AS Active_cases,
ROUND ((sum(p.numberOf_Active_Cases())/sum(p.no_of_covid_confirmed))*100,3) AS Active_Rate,
SUM(p. numberOf_CLOSED_CASES()) AS closed_cases,
ROUND (SUM(p. numberOf_CLOSED_CASES())/SUM(p.no_of_covid_confirmed)*100,4) AS CLOSED_CASES_RATE

FROM   panademic_info p

WHERE  p.pdate='21-MAR-2020' AND p.no_of_covid_confirmed>0 AND NOT p.cid.country_Region ='China'

GROUP by p.pdate;
```

The query results are displayed in a table with the following columns and values:

PDATE	CONFIRMED	DEATHS	MOTILITY_RATE	RECOVERED	RECOVERY_RATE	ACTIVE_CASES	ACTIVE_RATE	CLOSED_CASES	CLOSED_CASES_RATE
1 21-MAR-20	223219	9714	4.3518	19642	8.7994	193863	86.849	29356	13.1512

- 223219 have confirmed, 9714 people have died with a rate of 4.3518%, 19642 recovered with a rate of 8.7994%, and 193863 people are currently infected with a rate of 86.849 %so far from the coronavirus COVID-19 outbreak as of 21-MAR- 2020 in in outside of China.

Coronavirus COVID-19 Cases – Comparisons

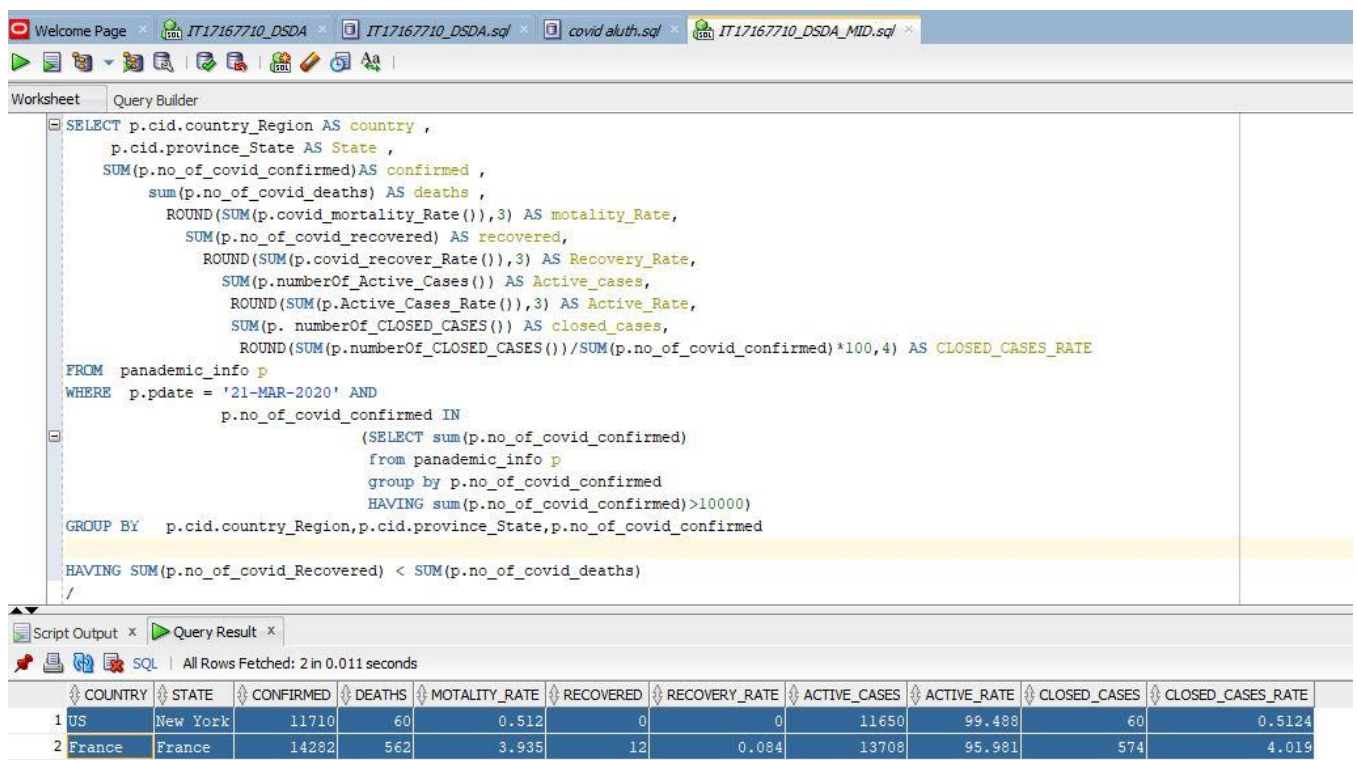
This report contains about full detailed report of the countries which exceeded 10000 patients of **COVID-19**. And each state of them where death toll is reported more than the recovery toll on 21th March,2020

```
SELECT p.cid.country_Region AS country ,
       p.cid.province_State AS State ,
       SUM(p.no_of_covid_confirmed) AS confirmed ,
       sum(p.no_of_covid_deaths) AS deaths ,
       ROUND (SUM(p.covid_mortality_Rate()),3) AS mortality_Rate,
       SUM(p.no_of_covid_recovered) AS recovered,
       ROUND (SUM(p.covid_recover_Rate()),3) AS Recovery_Rate,
       SUM(p.numberof_Active_Cases()) AS Active_cases,
       ROUND (SUM(p.Active_Cases_Rate()),3) AS Active_Rate,
       SUM(p. numberof_CLOSED_CASES()) AS closed_cases,

ROUND (SUM(p.numberof_CLOSED_CASES())/SUM(p.no_of_covid_confirmed)*100,4) AS
CLOSED_CASES_RATE
FROM panademic_info p
WHERE p.pdate = '21-MAR-2020' AND
      p.no_of_covid_confirmed IN
      (SELECT sum(p.no_of_covid_confirmed)
       from panademic_info p
       group by p.no_of_covid_confirmed
       HAVING sum(p.no_of_covid_confirmed)>10000)

GROUP BY
p.cid.country_Region,p.cid.province_State,p.no_of_covid_confirmed

HAVING SUM(p.no_of_covid_Recovered) < SUM(p.no_of_covid_deaths)
/
```



The screenshot shows a database query editor with the following tabs: Welcome Page, IT17167710_DSDA, IT17167710_DSDA.sql, covid_aluth.sql, and IT17167710_DSDA_MID.sql. The query is displayed in the main editor area, and the results are shown in the Query Result tab at the bottom.

COUNTRY	STATE	CONFIRMED	DEATHS	MORTALITY_RATE	RECOVERED	RECOVERY_RATE	ACTIVE_CASES	ACTIVE_RATE	CLOSED_CASES	CLOSED_CASES_RATE
US	New York	11710	60	0.512	0	0	11650	99.488	60	0.5124
France	France	14282	562	3.935	12	0.084	13708	95.981	574	4.019

8)

Here is the 2 SEQUENCE and the 2 Triggers that is used to create an incrementing numeric key, and to automatically populates the primary keys for 2 tables

```
SQL>
SQL> create SEQUENCE seq1
  2  start with 1
  3  increment by 1;

Sequence created.

SQL> CREATE OR REPLACE TRIGGER covid_countries_trigger
  2  BEFORE INSERT
  3  ON countries_tlb
  4  REFERENCING NEW AS NEW
  5  FOR EACH ROW
  6  BEGIN
  7  SELECT seq1.nextval INTO :NEW.countryid FROM dual;
  8  END;
  9  /

Trigger created.

SQL> create SEQUENCE seq2
  2  start with 1
  3  increment by 1;

Sequence created.

SQL>
SQL> CREATE OR REPLACE TRIGGER covid_panadamic_trigger
  2  BEFORE INSERT
  3  ON panademic_info
  4  REFERENCING NEW AS NEW
  5  FOR EACH ROW
  6  BEGIN
  7  SELECT seq2.nextval INTO :NEW.panademicID FROM dual;
  8  END;
  9  /

Trigger created.
```

after the CSV files load into the Oracle tables. I wrote simple query to verify whether the data loading done correctly

```
SELECT p.no_covid_confirmed ,p.no_of_covid_recovered,p.no_covid_deaths
FROM panademic_info p
WHERE p.cid.country_Region='Thailand' AND p.pdate='22-JAN-0020';
```

Query Result x | Script Output x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 1 in 0.031 seconds

NO_COVID_CONFIRMED	NO_OF_COVID_RECOVERED	NO_COVID_DEATHS
2	0	0

	A	B	C	D	E
1	Province/	Country/Region	Lat	Long	1/22/20
2		Thailand	15	101	2
3		Japan	36	138	2

covid_Confirmed.CSV'

	A	B	C	D	E
	Province/S	Country/R	Lat	Long	1/22/20
		Thailand	15	101	0
		Japan	36	138	0

covid_Death.CSV'

	A	B	C	D	E
	Province/	Country/R	Lat	Long	1/22/20
		Thailand	15	101	0
		Japan	36	138	0

covid_Recovered.CSV'

References

<https://alloraclesql.blogspot.com/2017/03/csv-file-loader-in-oracle-using-plsql.html> - load CSV

<https://stackoverflow.com/questions/11296361/how-to-create-id-with-auto-increment-on-oracle> -

<https://www.tekstream.com/resource-center/ora-01843-not-a-valid-month/> - Date format

<http://nimishgarg.blogspot.com/2013/04/load-csv-file-in-oracle-using-plsql.html> - load CSV