JDBC VS ORM EXPLORATORY ANALYSIS

IMPORTING PACKAGES

import mysql.connector
import pandas as pd
import numpy as np
import math
import statistics as stat
import matplotlib.pyplot as plt
import seaborn as sns
import random
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
from getpass import getpass
from time import time

IMPORTING PACKAGES

```
import mysql.connector
import pandas as pd
import numpy as np
import math
import statistics as stat
import matplotlib.pyplot as plt
import seaborn as sns
import random
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
from getpass import getpass
from time import time
```

CREATING LISTS TO STORE THE EXECUTION TIME JDBC AND ORM

time_taken_jdbc_with_parition=[]
time_taken_jdbc_without_parition=[]
time_taken_orm_with_parition=[]
time_taken_orm_without_parition=[]

CREATING LISTS TO STORE THE EXECUTION TIME

```
time_taken_jdbc_with_parition=[]
time_taken_jdbc_without_parition=[]
time_taken_orm_with_parition=[]
time_taken_orm_without_parition=[]
```

JDBC CONNECTION

Enter the username and Password to connect to the MYSQL DB:

```
print(" Connecting to the localhost ")
global usrnm;
usrnm=input("Enter username to connect to the MYSQL DB: ")
global pwd;
pwd=getpass("Enter password  to connect to the MYSQL DB: ")
```

Enter the username and Password to connect to the MYSQL DB:

```
print(" Connecting to the localhost ")
global usrnm;
usrnm=input("Enter username to connect to the MYSQL DB: ")
global pwd;
pwd=getpass("Enter password  to connect to the MYSQL DB: ")
```

```
 Connecting to the localhost
Enter username to connect to the MYSQL DB: root
Enter password  to connect to the MYSQL DB: ·····
```

JDBC - WITH PARTITION

SCENARIO 1:

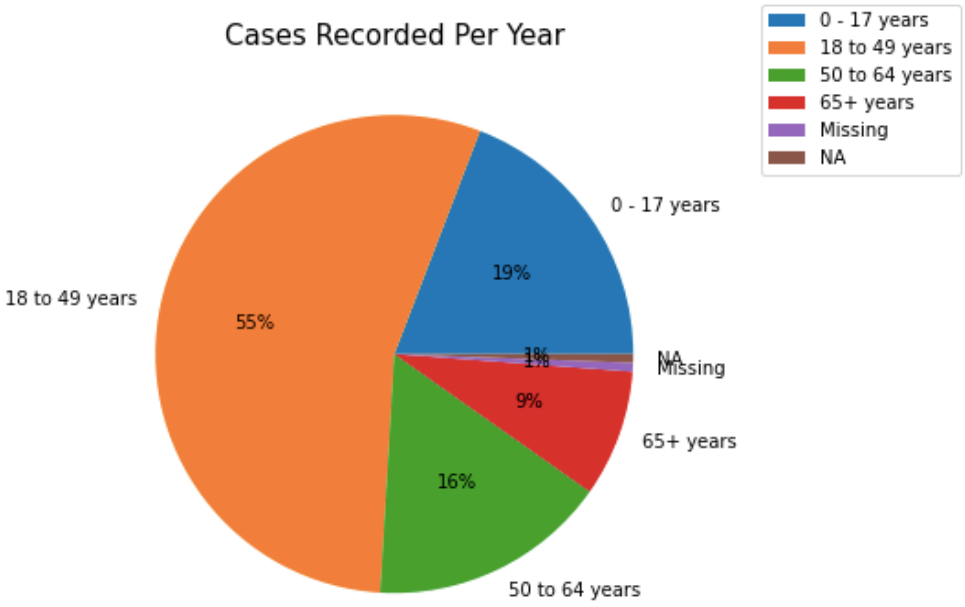
Display the number of cases recorded per age group:

```
from mysql.connector import connect, Error
try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start1=time()
sql1 = """\
select age_group,count(*) as 'Case_recorded_per_age_group'
from group_project_225.cdc_covid_data_local group by age_group order by count(*) desc;"""
with connection.cursor() as cur1:
cur1.execute(sql1)
Cases_Recorded_Per_Age_Group_JDBC_With_Partition=pd.DataFrame (cur1. fetchall ())
end1=time()
tt1=end1-start1
time_taken_jdbc_with_parition.append(tt1)
except Error as e:
print(e)

Cases_Recorded_Per_Age_Group_JDBC_With_Partition.columns=['Age_Group','Cases_Recorded_Per_Age
_Group']
Cases_Recorded_Per_Age_Group_JDBC_With_Partition
```

SCENARIO 1 : Display the number of cases recorded per age group

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start1=time()
        sql1 = """\
        select age_group,count(*) as 'Case_recorded_per_age_group'
        from group_project_225.cdc_covid_data_local group by age_group order by count(*) desc;"""
        with connection.cursor() as cur1:
            cur1.execute(sql1)
            Cases_Recorded_Per_Age_Group_JDBC_With_Partition=pd.DataFrame (cur1. fetchall ())
        end1=time()
        tt1=end1-start1
        time_taken_jdbc_with_parition.append(tt1)
except Error as e:
    print(e)

Cases_Recorded_Per_Age_Group_JDBC_With_Partition.columns =['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_JDBC_With_Partition
```

	Age_Group	Cases_Recorded_Per_Age_Group
0	18 to 49 years	1862627
1	0 - 17 years	649829
2	50 to 64 years	548659
3	65+ years	291220
4	Missing	20175
5	NA	19740



SCENARIO 2:

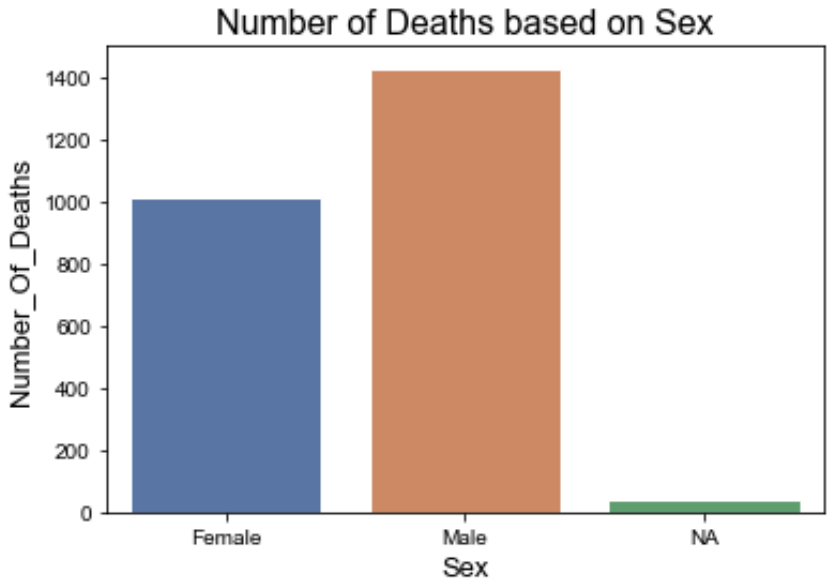
Display the number of deaths based on Sex:

```
from mysql.connector import connect, Error

try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start2=time()
sql2 = """\select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_local group by death_yn,
sex having death_yn='Yes'"""
with connection.cursor() as cur2:
cur2.execute(sql2)
Death_Per_Gender_JDBC_With_Partition=pd.DataFrame(cur2. fetchall ())
end2=time()
tt2=end2-start2
time_taken_jdbc_with_parition.append(tt2)
except Error as e:
print(e)

Death_Per_Gender_JDBC_With_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_JDBC_With_Partition
```

SCENARIO 2 : Display the number of deaths based on Sex

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start2=time()
        sql2 = """\select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_local group by death_yn,
        sex having death_yn='Yes'"""
        with connection.cursor() as cur2:
            cur2.execute(sql2)
            Death_Per_Gender_JDBC_With_Partition=pd.DataFrame(cur2. fetchall ())
        end2=time()
        tt2=end2-start2
        time_taken_jdbc_with_parition.append(tt2)
except Error as e:
    print(e)

Death_Per_Gender_JDBC_With_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_JDBC_With_Partition
```

	Death_YesOrNo	Sex	Number_Of_Deaths
0	Yes	Female	1009
1	Yes	Male	1428
2	Yes	NA	40



SCENARIO 3:

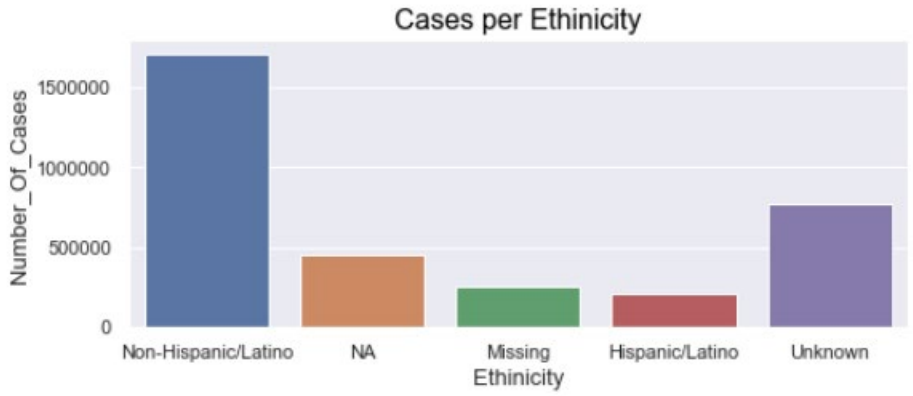
Display the number of cases per ethnicity

```
from mysql.connector import connect, Error

try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start3=time()
sql3 = """\
select ethnicity,count(*) as number_of_cases from group_project_225.cdc_covid_data_local
group by ethnicity;"""
with connection.cursor() as cur3:
cur3.execute(sql3)
Cases_Per_Ethinicity_JDBC_With_Partition=pd.DataFrame(cur3. fetchall ())
end3=time()
tt3=end3-start3
time_taken_jdbc_with_parition.append(tt3)
except Error as e:
print(e)

Cases_Per_Ethinicity_JDBC_With_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_JDBC_With_Partition
```

SCENARIO 3 : Display the number of cases per ethinicity

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start3=time()
        sql3 = """\
        select ethnicity,count(*) as number_of_cases from group_project_225.cdc_covid_data_local
        group by ethnicity;"""
        with connection.cursor() as cur3:
            cur3.execute(sql3)
            Cases_Per_Ethinicity_JDBC_With_Partition=pd.DataFrame(cur3. fetchall ())
        end3=time()
        tt3=end3-start3
        time_taken_jdbc_with_parition.append(tt3)
except Error as e:
    print(e)

Cases_Per_Ethinicity_JDBC_With_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_JDBC_With_Partition
```

	Ethinicity	Number_Of_Cases
0	Non-Hispanic/Latino	1712966
1	NA	455215
2	Missing	247276
3	Hispanic/Latino	205152
4	Unknown	771641



SCENARIO 4:

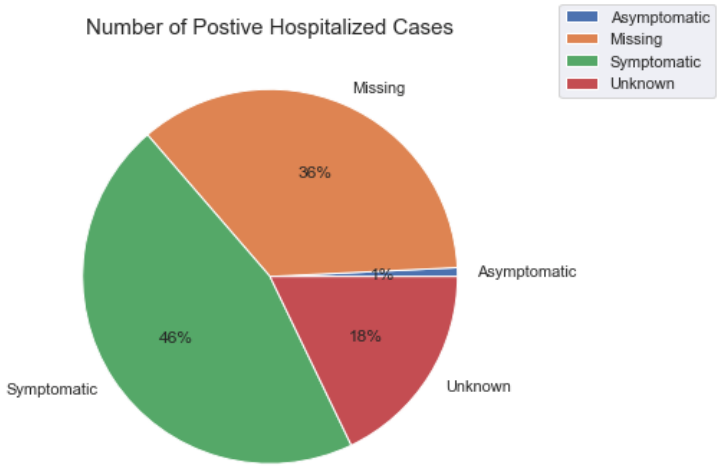
Number of Positive Hospitalized Cases:

```
from mysql.connector import connect, Error

try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start4=time()
sql4 = """\
select symptom_status,count(*) from group_project_225.cdc_covid_data_local
group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;"""
with connection.cursor() as cur4:
cur4.execute(sql4)
Hosp_Positive_Cases_JDBC_With_Partition=pd.DataFrame(cur4. fetchall ())
end4=time()
tt4=end4-start4
time_taken_jdbc_with_parition.append(tt4)
except Error as e:
print(e)

Hosp_Positive_Cases_JDBC_With_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_JDBC_With_Partition
```

SCENARIO 4 : Number of Positive Hospitalized Cases

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start4=time()
        sql4 = """\
        select symptom_status,count(*) from group_project_225.cdc_covid_data_local
        group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;"""
        with connection.cursor() as cur4:
            cur4.execute(sql4)
            Hosp_Positive_Cases_JDBC_With_Partition=pd.DataFrame(cur4. fetchall ())
        end4=time()
        tt4=end4-start4
        time_taken_jdbc_with_parition.append(tt4)
except Error as e:
    print(e)

Hosp_Positive_Cases_JDBC_With_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_JDBC_With_Partition
```

	Symptom_Status	Number_Of_Cases
0	Asymptomatic	492
1	Missing	22990
2	Symptomatic	29596
3	Unknown	11544



SCENARIO 5:

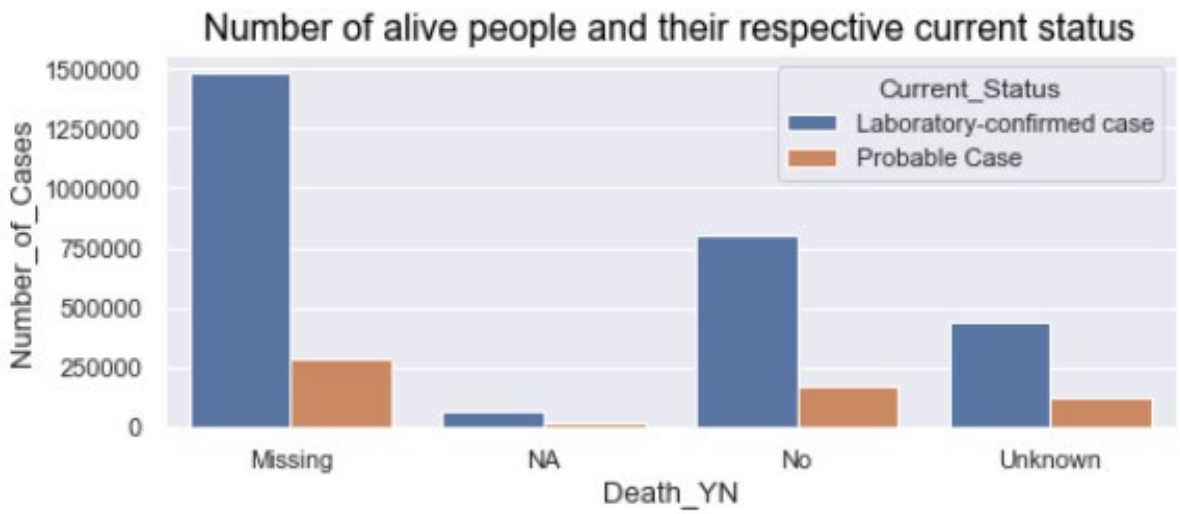
Number of cases and respective current status of alive people:

```
from mysql.connector import connect, Error

try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start5=time()
sql5 = """\
select death_yn,current_status,count(*) as 'number_people_died' from
group_project_225.cdc_covid_data_local group by death_yn,current_status
having death_yn <> 'Yes' order by death_yn,current_status;"""
with connection.cursor() as cur5:
cur5.execute(sql5)
Alive_Status_JDBC_With_Partition=pd.DataFrame(cur5. fetchall ())
end5=time()
tt5=end5-start5
time_taken_jdbc_with_parition.append(tt5)
except Error as e:
print(e)

Alive_Status_JDBC_With_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_JDBC_With_Partition
```

SCENARIO 5 : Number of cases and respective current_status of alive people

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start5=time()
        sql5 = """\
           select death_yn,current_status,count(*) as 'number_people_died' from
           group_project_225.cdc_covid_data_local group by death_yn,current_status
           having death_yn <> 'Yes' order by death_yn,current_status;"""
        with connection.cursor() as cur5:
            cur5.execute(sql5)
            Alive_Status_JDBC_With_Partition=pd.DataFrame(cur5. fetchall ())
        end5=time()
        tt5=end5-start5
        time_taken_jdbc_with_parition.append(tt5)
except Error as e:
    print(e)

Alive_Status_JDBC_With_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_JDBC_With_Partition
```

	Death_YN	Current_Status	Number_of_Cases
0	Missing	Laboratory-confirmed case	1486202
1	Missing	Probable Case	283238
2	NA	Laboratory-confirmed case	66297
3	NA	Probable Case	19494
4	No	Laboratory-confirmed case	803349
5	No	Probable Case	174811
6	Unknown	Laboratory-confirmed case	437111
7	Unknown	Probable Case	119271



JDBC - WITHOUT PARTITION

SCENARIO 1:

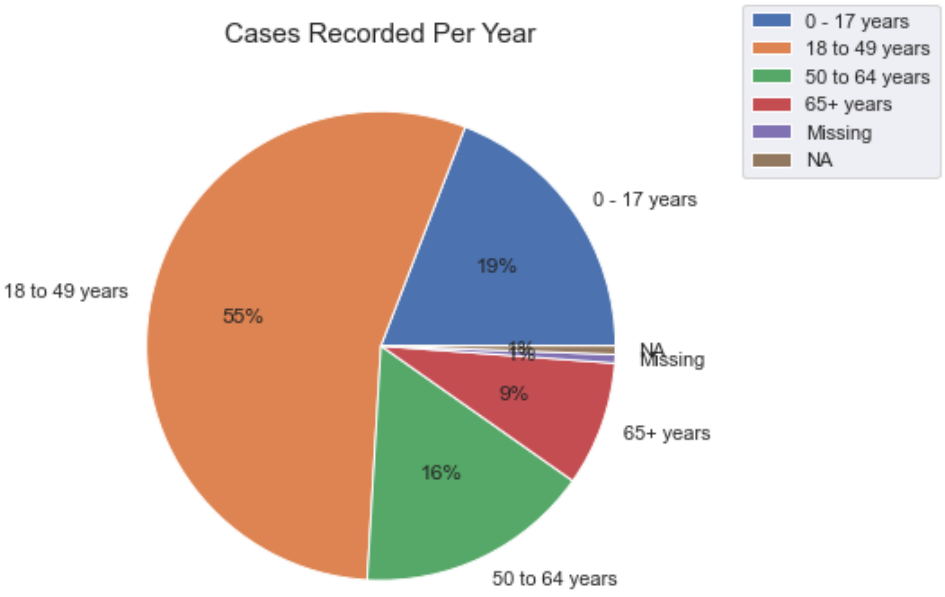
Display the number of cases recorded per age group:

```
from mysql.connector import connect, Error
try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start1=time()
sql1 = """select age_group,count(*) as 'Case_recorded_per_age_group'
from group_project_225.cdc_covid_data_without_partition group by age_group order by count(*)
desc;"""
with connection.cursor() as cur1:
cur1.execute(sql1)
Cases_Recorded_Per_Age_Group_JDBC_Without_Partition=pd.DataFrame (cur1. fetchall ())
end1=time()
tt1=end1-start1
time_taken_jdbc_without_parition.append(tt1)
except Error as e:
print(e)
Cases_Recorded_Per_Age_Group_JDBC_Without_Partition.columns
=['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_JDBC_Without_Partition
```

SCENARIO 1 : Display the number of cases recorded per age group `

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start1=time()
        sql1 = """select age_group,count(*) as 'Case_recorded_per_age_group'
        from group_project_225.cdc_covid_data_without_partition group by age_group order by count(*) desc;"""
        with connection.cursor() as cur1:
            cur1.execute(sql1)
            Cases_Recorded_Per_Age_Group_JDBC_Without_Partition=pd.DataFrame (cur1. fetchall ())
        end1=time()
        tt1=end1-start1
        time_taken_jdbc_without_parition.append(tt1)
except Error as e:
    print(e)

Cases_Recorded_Per_Age_Group_JDBC_Without_Partition.columns =['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_JDBC_Without_Partition
```

	Age_Group	Cases_Recorded_Per_Age_Group
0	18 to 49 years	1862627
1	0 - 17 years	649829
2	50 to 64 years	548659
3	65+ years	291220
4	Missing	20175
5	NA	19740



SCENARIO 2:

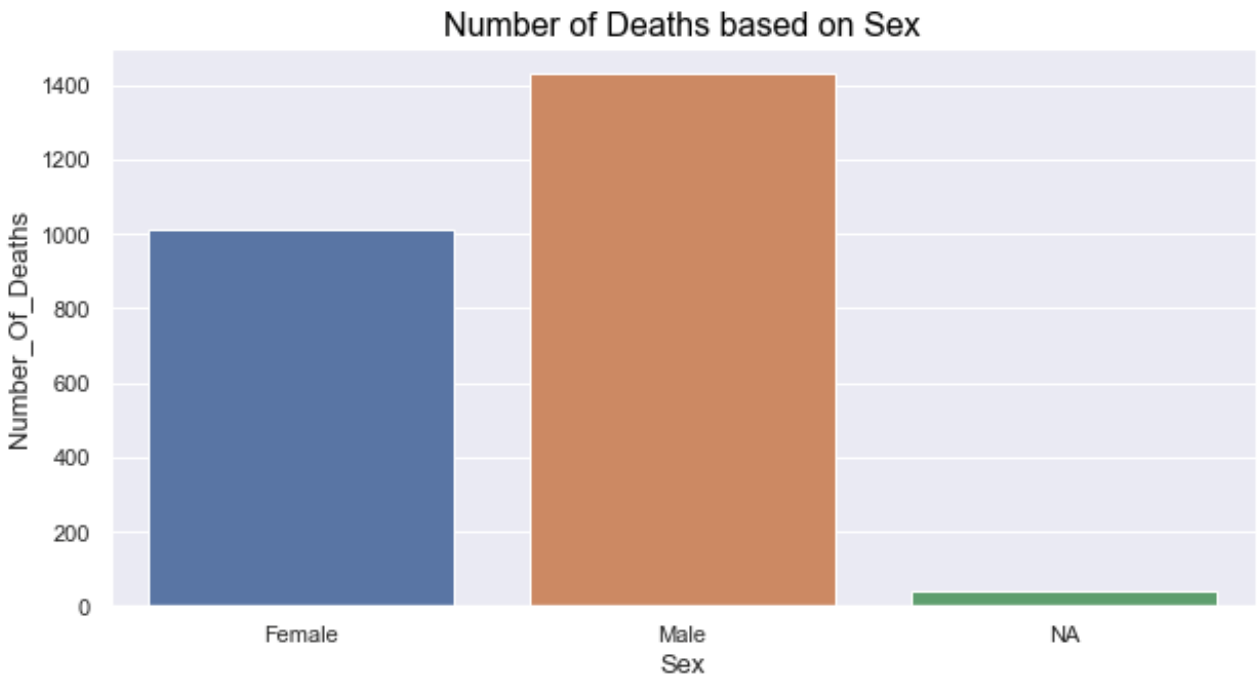
Display the number of deaths based on Sex:

```
from mysql.connector import connect, Error

try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start2=time()
sql2 = """select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_without_partition
group by death_yn,sex having death_yn='Yes'"""
with connection.cursor() as cur2:
cur2.execute(sql2)
Death_Per_Gender_JDBC_Without_Partition=pd.DataFrame(cur2. fetchall ())
end2=time()
tt2=end2-start2
time_taken_jdbc_without_parition.append(tt2)
except Error as e:
print(e)

Death_Per_Gender_JDBC_Without_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_JDBC_Without_Partition
```

SCENARIO 2 : Display the number of deaths based on Sex

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start2=time()
        sql2 = """select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_without_partition
        group by death_yn,sex having death_yn='Yes'"""
        with connection.cursor() as cur2:
            cur2.execute(sql2)
            Death_Per_Gender_JDBC_Without_Partition=pd.DataFrame(cur2. fetchall ())
        end2=time()
        tt2=end2-start2
        time_taken_jdbc_without_parition.append(tt2)
except Error as e:
    print(e)

Death_Per_Gender_JDBC_Without_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_JDBC_Without_Partition
```

	Death_YesOrNo	Sex	Number_Of_Deaths
0	Yes	Female	1009
1	Yes	Male	1428
2	Yes	NA	40



SCENARIO 3:

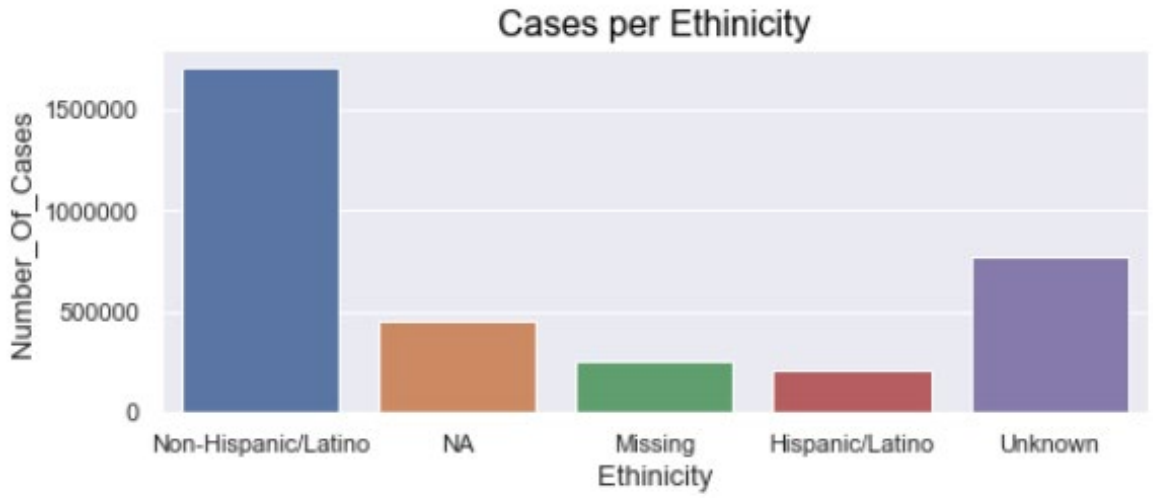
Display the number of cases per ethinicity:

```
from mysql.connector import connect, Error

try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start3=time()
sql3 = """select ethnicity,count(*) as number_of_cases from
group_project_225.cdc_covid_data_without_partition group by ethnicity;"""
with connection.cursor() as cur3:
cur3.execute(sql3)
Cases_Per_Ethinicity_JDBC_Without_Partition=pd.DataFrame(cur3. fetchall ())
end3=time()
tt3=end3-start3
time_taken_jdbc_without_parition.append(tt3)
except Error as e:
print(e)

Cases_Per_Ethinicity_JDBC_Without_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_JDBC_Without_Partition
```

SCENARIO 2 : Display the number of deaths based on Sex

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start2=time()
        sql2 = """select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_without_partition
        group by death_yn,sex having death_yn='Yes'"""
        with connection.cursor() as cur2:
            cur2.execute(sql2)
            Death_Per_Gender_JDBC_Without_Partition=pd.DataFrame(cur2. fetchall ())
        end2=time()
        tt2=end2-start2
        time_taken_jdbc_without_parition.append(tt2)
except Error as e:
    print(e)

Death_Per_Gender_JDBC_Without_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_JDBC_Without_Partition
```

	Death_YesOrNo	Sex	Number_Of_Deaths
0	Yes	Female	1009
1	Yes	Male	1428
2	Yes	NA	40



SCENARIO 4:

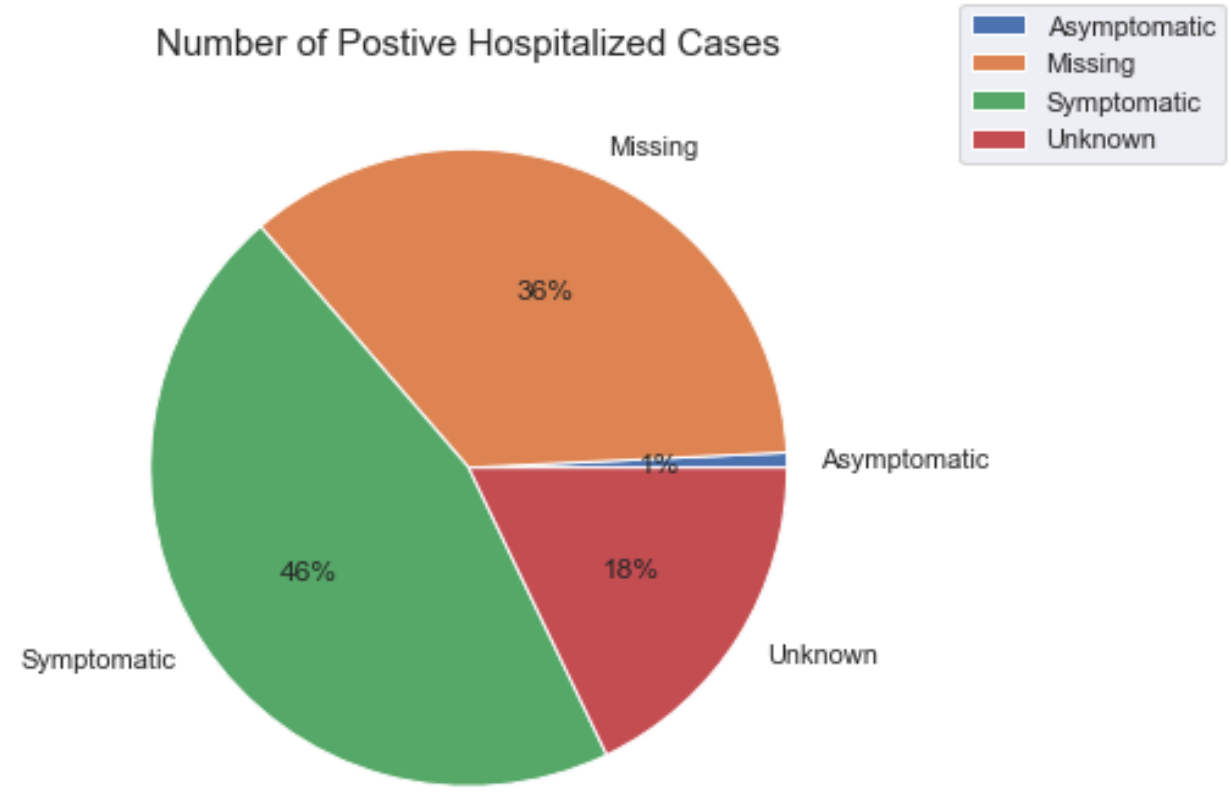
Number of Positive Hospitalized Cases:

```
from mysql.connector import connect, Error


try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start4=time()
sql4 = """select symptom_status,count(*) from group_project_225.cdc_covid_data_without_partition
group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;"""
with connection.cursor() as cur4:
cur4.execute(sql4)
Hosp_Positive_Cases_JDBC_Without_Partition=pd.DataFrame(cur4. fetchall ())
end4=time()
tt4=end4-start4
time_taken_jdbc_without_parition.append(tt4)
except Error as e:
print(e)

Hosp_Positive_Cases_JDBC_Without_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_JDBC_Without_Partition
```

SCENARIO 4 : Number of Positive Hospitalized Cases

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start4=time()
        sql4 = """select symptom_status,count(*) from group_project_225.cdc_covid_data_without_partition
        group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;"""
        with connection.cursor() as cur4:
            cur4.execute(sql4)
            Hosp_Positive_Cases_JDBC_Without_Partition=pd.DataFrame(cur4. fetchall ())
        end4=time()
        tt4=end4-start4
        time_taken_jdbc_without_parition.append(tt4)
except Error as e:
    print(e)

Hosp_Positive_Cases_JDBC_Without_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_JDBC_Without_Partition
```

	Symptom_Status	Number_Of_Cases
0	Asymptomatic	492
1	Missing	22990
2	Symptomatic	29596
3	Unknown	11544



SCENARIO 5:

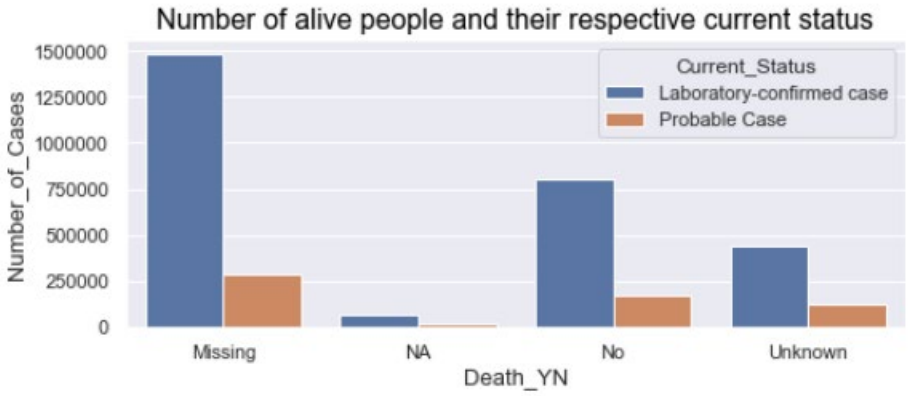
Number of cases and respective current status of alive people:

```
from mysql.connector import connect, Error
try:
with connect(host="localhost",user=usrnm, password=pwd
) as connection:
start5=time()
sql5 = """select death_yn,current_status,count(*) as 'number_people_died' from
group_project_225.cdc_covid_data_without_partition group by death_yn,current_status
having death_yn <> 'Yes' order by death_yn,current_status;"""
with connection.cursor() as cur5:
cur5.execute(sql5)
Alive_Status_JDBC_Without_Partition=pd.DataFrame(cur5. fetchall ())
end5=time()
tt5=end5-start5
time_taken_jdbc_without_parition.append(tt5)
except Error as e:
print(e)

Alive_Status_JDBC_Without_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_JDBC_Without_Partition
```

SCENARIO 5 : Number of cases and respective current status of alive people

```
from mysql.connector import connect, Error

try:
     with connect(host="localhost",user=usrnm, password=pwd
    ) as connection:
        start5=time()
        sql5 = """select death_yn,current_status,count(*) as 'number_people_died' from
        group_project_225.cdc_covid_data_without_partition group by death_yn,current_status
        having death_yn <> 'Yes' order by death_yn,current_status;"""
        with connection.cursor() as cur5:
            cur5.execute(sql5)
            Alive_Status_JDBC_Without_Partition=pd.DataFrame(cur5. fetchall ())
        end5=time()
        tt5=end5-start5
        time_taken_jdbc_without_parition.append(tt5)
except Error as e:
    print(e)

Alive_Status_JDBC_Without_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_JDBC_Without_Partition
```

	Death_YN	Current_Status	Number_of_Cases
0	Missing	Laboratory-confirmed case	1486202
1	Missing	Probable Case	283238
2	NA	Laboratory-confirmed case	66297
3	NA	Probable Case	19494
4	No	Laboratory-confirmed case	803349
5	No	Probable Case	174811
6	Unknown	Laboratory-confirmed case	437111
7	Unknown	Probable Case	119271



JDBC-WITH PARTITION VS WITHOUT PARTITION EXECUTION TIME

	Query_Num	JDBC_With_Partition_Time	JDBC_Without_Partition_Time
0	1	4.634861	3.823150
1	2	6.623851	8.117040
2	3	3.551446	4.151696
3	4	4.368735	5.124085
4	5	4.430888	5.157737

ORM CONNECTION TO MYSQL DB USING SQL ALCHEMY

```
from sqlalchemy import Column, Integer, Text
from sqlalchemy.dialects.postgresql import JSON, JSONB
import sqlalchemy as db

engine = db.create_engine('mysql://root:GaYu6793@localhost:3306/group_project_225')
connection1 = engine.connect()
metadata = db.MetaData()

covid_data_partition = db.Table('cdc_covid_data_local', metadata, autoload=True,
autoload_with=engine)
```


```
from sqlalchemy import Column, Integer, Text
from sqlalchemy.dialects.postgresql import JSON, JSONB
import sqlalchemy as db
engine = db.create_engine('mysql://root:GaYu6793@localhost:3306/group_project_225')
connection1 = engine.connect()
metadata = db.MetaData()
covid_data_partition = db.Table('cdc_covid_data_local', metadata, autoload=True, autoload_with=engine)
```

ORM – WITH PARTITION

SCENARIO 1:

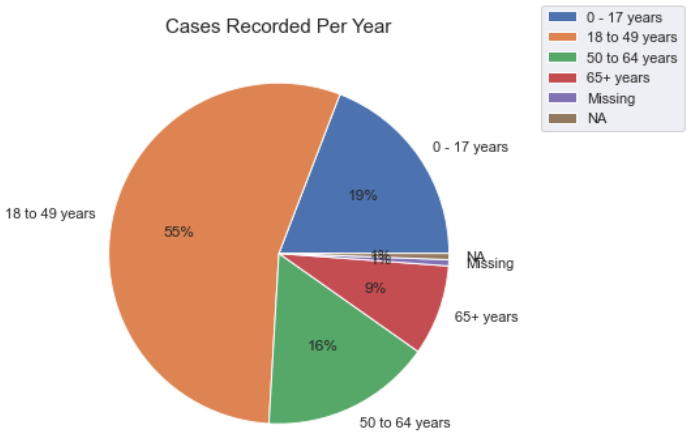
Display the number of cases recorded per age group:

```
from sqlalchemy import text
start6=time()
sql6 = text("""select age_group,count(*) as 'Case_recorded_per_age_group' from
group_project_225.cdc_covid_data_local group by age_group order by count(*) desc;""")
result6 = engine.execute(sql6)
end6=time()
tt6=end6-start6
time_taken_orm_with_parition.append(tt6)
Cases_Recorded_Per_Age_Group_ORM_With_Partition=pd.DataFrame(result6)
Cases_Recorded_Per_Age_Group_ORM_With_Partition.columns
=['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_ORM_With_Partition
```

SCENARIO 1 : Display the number of cases recorded per age group

```
from sqlalchemy import text
start6=time()
sql6 = text("""select age_group,count(*) as 'Case_recorded_per_age_group' from
            group_project_225.cdc_covid_data_local group by age_group order by count(*) desc;""")
result6 = engine.execute(sql6)
end6=time()
tt6=end6-start6
time_taken_orm_with_parition.append(tt6)
Cases_Recorded_Per_Age_Group_ORM_With_Partition=pd.DataFrame(result6)


Cases_Recorded_Per_Age_Group_ORM_With_Partition.columns =['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_ORM_With_Partition
```

	Age_Group	Cases_Recorded_Per_Age_Group
0	18 to 49 years	1862627
1	0 - 17 years	649829
2	50 to 64 years	548659
3	65+ years	291220
4	Missing	20175
5	NA	19740



SCENARIO 2:

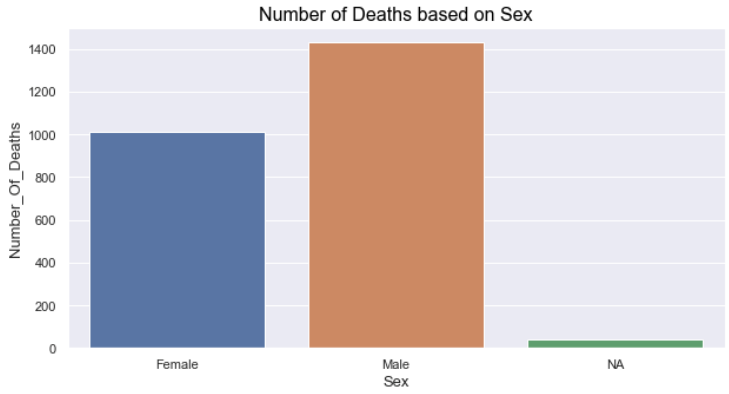
Display the number of deaths based on Sex:

```
from sqlalchemy import text
start7=time()
sql7 = text("""select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_local
group by death_yn,sex having death_yn='Yes';""")
result7 = engine.execute(sql7)
end7=time()
tt7=end7-start7
time_taken_orm_with_parition.append(tt7)
Death_Per_Gender_ORM_With_Partition=pd.DataFrame(result7)

Death_Per_Gender_ORM_With_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_ORM_With_Partition
```

SCENARIO 2 : Display the number of deaths based on Sex

```
from sqlalchemy import text
start7=time()
sql7 = text("""select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_local
            group by death_yn,sex having death_yn='Yes';""")
result7 = engine.execute(sql7)
end7=time()
tt7=end7-start7
time_taken_orm_with_parition.append(tt7)
Death_Per_Gender_ORM_With_Partition=pd.DataFrame(result7)


Death_Per_Gender_ORM_With_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_ORM_With_Partition
```

	Death_YesOrNo	Sex	Number_Of_Deaths
0	Yes	Female	1009
1	Yes	Male	1428
2	Yes	NA	40



SCENARIO 3:

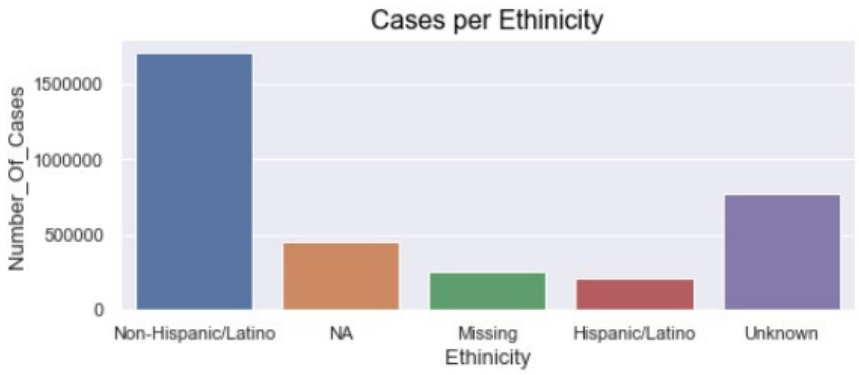
Display the number of cases per ethnicity:

```
from sqlalchemy import text
start8=time()
sql8 = text("""select ethnicity,count(*) as number_of_cases from
group_project_225.cdc_covid_data_local  group by ethnicity;""")

result8 = engine.execute(sql8)
end8=time()
tt8=end8-start8
time_taken_orm_with_parition.append(tt8)
Cases_Per_Ethinicity_ORM_With_Partition=pd.DataFrame(result8)


Cases_Per_Ethinicity_ORM_With_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_ORM_With_Partition
```

SCENARIO 3 : Display the number of cases per ethinicity

```
from sqlalchemy import text
start8=time()
sql8 = text("""select ethnicity,count(*) as number_of_cases from group_project_225.cdc_covid_data_local
               group by ethnicity;""")
result8 = engine.execute(sql8)
end8=time()
tt8=end8-start8
time_taken_orm_with_parition.append(tt8)
Cases_Per_Ethinicity_ORM_With_Partition=pd.DataFrame(result8)


Cases_Per_Ethinicity_ORM_With_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_ORM_With_Partition
```

	Ethinicity	Number_Of_Cases
0	Non-Hispanic/Latino	1712966
1	NA	455215
2	Missing	247276
3	Hispanic/Latino	205152
4	Unknown	771641



SCENARIO 4:

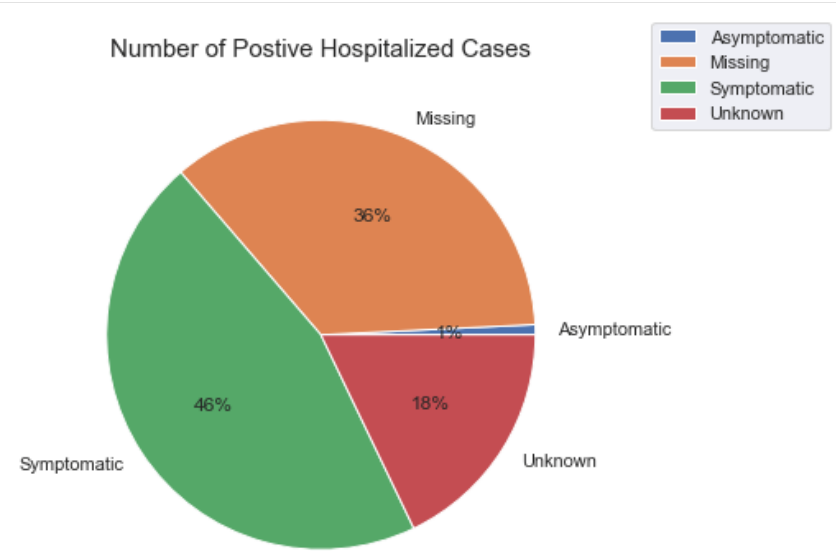
Number of Positive Hospitalized Cases:

```
from sqlalchemy import text
start9=time()
sql9 = text("""select symptom_status,count(*) from group_project_225.cdc_covid_data_local
group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;""")
result9 = engine.execute(sql9)
end9=time()
tt9=end9-start9
time_taken_orm_with_parition.append(tt9)
Hosp_Positive_Cases_ORM_With_Partition=pd.DataFrame(result9)

Hosp_Positive_Cases_ORM_With_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_ORM_With_Partition
```

SCENARIO 4 : Number of Positive Hospitalized Cases

```
from sqlalchemy import text
start9=time()
sql9 = text("""select symptom_status,count(*) from group_project_225.cdc_covid_data_local
            group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;""")
result9 = engine.execute(sql9)
end9=time()
tt9=end9-start9
time_taken_orm_with_parition.append(tt9)
Hosp_Positive_Cases_ORM_With_Partition=pd.DataFrame(result9)


Hosp_Positive_Cases_ORM_With_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_ORM_With_Partition
```

	Symptom_Status	Number_Of_Cases
0	Asymptomatic	492
1	Missing	22990
2	Symptomatic	29596
3	Unknown	11544



SCENARIO 5:

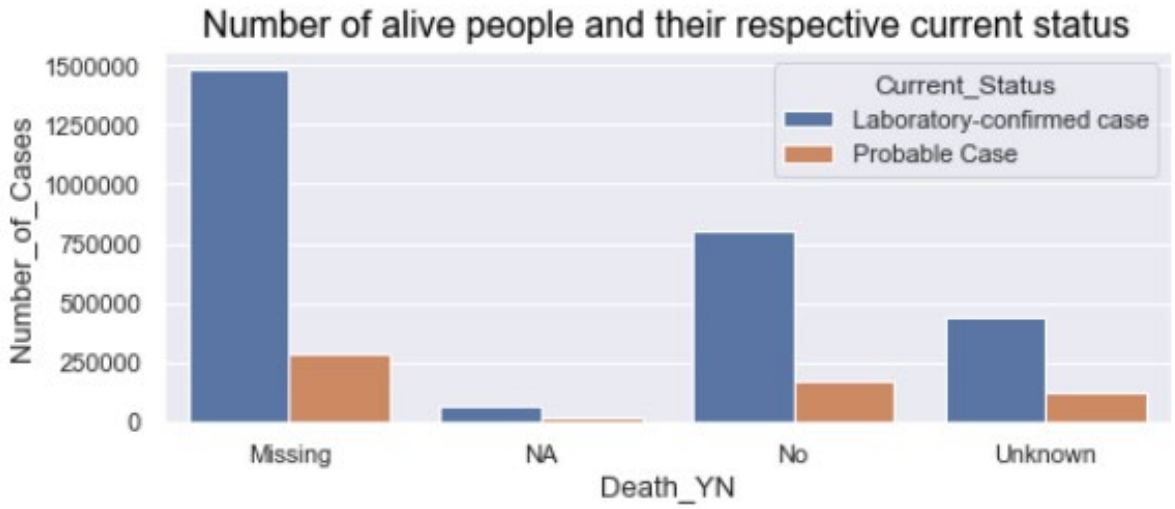
Number of cases and respective current status of alive people:

```
from sqlalchemy import text
start10=time()
sql10 = text("""select death_yn,current_status,count(*) as 'number_people_died' from
group_project_225.cdc_covid_data_local group by death_yn,current_status
having death_yn <> 'Yes' order by death_yn,current_status;""")
result10 = engine.execute(sql10)
end10=time()
tt10=end10-start10
time_taken_orm_with_parition.append(tt10)
Alive_Status_ORM_With_Partition=pd.DataFrame(result10)
Alive_Status_ORM_With_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_ORM_With_Partition
```

SCENARIO 5 : Number of cases and respective current_status of alive people

```
from sqlalchemy import text
start10=time()
sql10 = text("""select death_yn,current_status,count(*) as 'number_people_died' from
                group_project_225.cdc_covid_data_local group by death_yn,current_status
                having death_yn <> 'Yes' order by death_yn,current_status;""")
result10 = engine.execute(sql10)
end10=time()
tt10=end10-start10
time_taken_orm_with_parition.append(tt10)
Alive_Status_ORM_With_Partition=pd.DataFrame(result10)


Alive_Status_ORM_With_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_ORM_With_Partition
```

	Death_YN	Current_Status	Number_of_Cases
0	Missing	Laboratory-confirmed case	1486202
1	Missing	Probable Case	283238
2	NA	Laboratory-confirmed case	66297
3	NA	Probable Case	19494
4	No	Laboratory-confirmed case	803349
5	No	Probable Case	174811
6	Unknown	Laboratory-confirmed case	437111
7	Unknown	Probable Case	119271



ORM – WITHOUT PARTITION

SCENARIO 1:

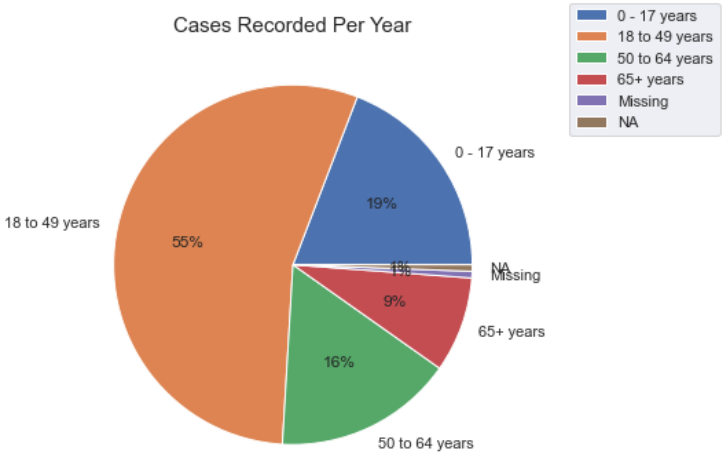
Display the number of cases recorded per age group:

```
from sqlalchemy import text
start11=time()
sql11 = text("""select age_group,count(*) as 'Case_recorded_per_age_group'
from group_project_225.cdc_covid_data_without_partition group by age_group
order by count(*) desc;""")
result11 = engine.execute(sql11)
end11=time()
tt11=end11-start11
time_taken_orm_without_parition.append(tt11)
Cases_Recorded_Per_Age_Group_ORM_Without_Partition=pd.DataFrame(result11)
Cases_Recorded_Per_Age_Group_ORM_Without_Partition.columns
=['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_ORM_Without_Partition
```

SCENARIO 1 : Display the number of cases recorded per age group

```
from sqlalchemy import text
start11=time()
sql11 = text("""select age_group,count(*) as 'Case_recorded_per_age_group'
                from group_project_225.cdc_covid_data_without_partition group by age_group
                order by count(*) desc;""")
result11 = engine.execute(sql11)
end11=time()
tt11=end11-start11
time_taken_orm_without_parition.append(tt11)
Cases_Recorded_Per_Age_Group_ORM_Without_Partition=pd.DataFrame(result11)


Cases_Recorded_Per_Age_Group_ORM_Without_Partition.columns =['Age_Group','Cases_Recorded_Per_Age_Group']
Cases_Recorded_Per_Age_Group_ORM_Without_Partition
```

	Age_Group	Cases_Recorded_Per_Age_Group
0	18 to 49 years	1862627
1	0 - 17 years	649829
2	50 to 64 years	548659
3	65+ years	291220
4	Missing	20175
5	NA	19740



SCENARIO 2:

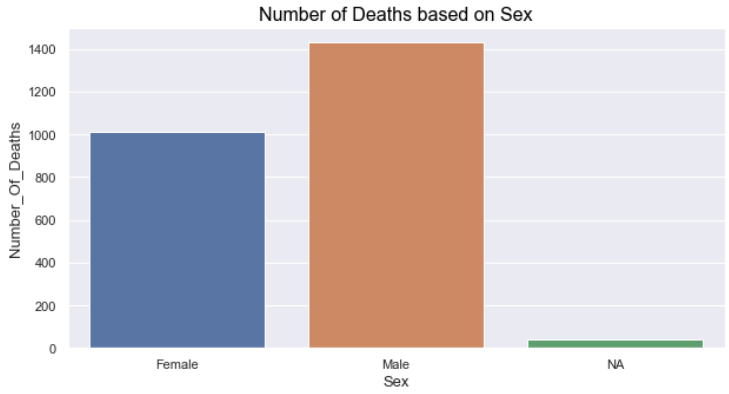
Display the number of deaths based on Sex:

```
from sqlalchemy import text
start12=time()
sql12 = text("""select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_without_partition
group by death_yn,sex having death_yn='Yes';""")
result12 = engine.execute(sql12)
end12=time()
tt12=end12-start12
time_taken_orm_without_parition.append(tt12)
Death_Per_Gender_ORM_Without_Partition=pd.DataFrame(result12)
Death_Per_Gender_ORM_Without_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_ORM_Without_Partition
```

SCENARIO 2 : Display the number of deaths based on Sex

```
from sqlalchemy import text
start12=time()
sql12 = text("""select death_yn,sex,count(*) from  group_project_225.cdc_covid_data_without_partition
                group by death_yn,sex having death_yn='Yes';""")
result12 = engine.execute(sql12)
end12=time()
tt12=end12-start12
time_taken_orm_without_parition.append(tt12)
Death_Per_Gender_ORM_Without_Partition=pd.DataFrame(result12)


Death_Per_Gender_ORM_Without_Partition.columns =['Death_YesOrNo','Sex','Number_Of_Deaths']
Death_Per_Gender_ORM_Without_Partition
```

	Death_YesOrNo	Sex	Number_Of_Deaths
0	Yes	Female	1009
1	Yes	Male	1428
2	Yes	NA	40



SCENARIO 3:

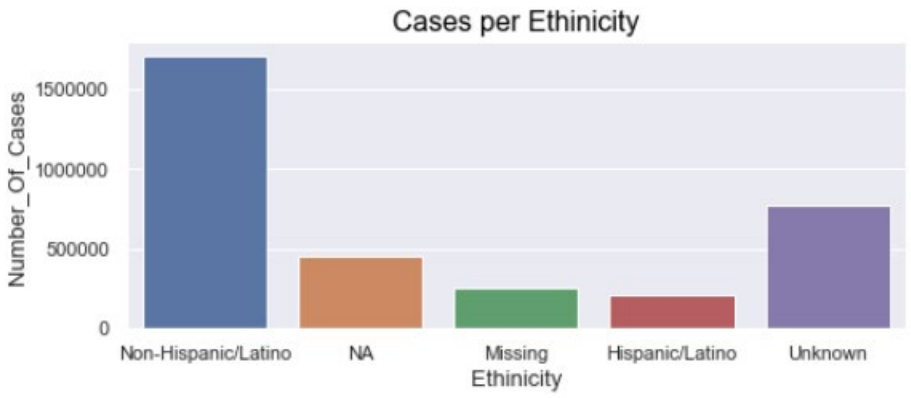
Display the number of cases per ethnicity:

```
from sqlalchemy import text
start13=time()
sql13 = text("""select ethnicity,count(*) as number_of_cases from
group_project_225.cdc_covid_data_without_partition group by ethnicity;""")
result13 = engine.execute(sql13)
end13=time()
tt13=end13-start13
time_taken_orm_without_parition.append(tt13)
Cases_Per_Ethinicity_ORM_Without_Partition=pd.DataFrame(result13)
Cases_Per_Ethinicity_ORM_Without_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_ORM_Without_Partition
```

SCENARIO 3 : Display the number of cases per ethinicity

```
from sqlalchemy import text
start13=time()
sql13 = text("""select ethnicity,count(*) as number_of_cases from
                group_project_225.cdc_covid_data_without_partition group by ethnicity;""")
result13 = engine.execute(sql13)
end13=time()
tt13=end13-start13
time_taken_orm_without_parition.append(tt13)
Cases_Per_Ethinicity_ORM_Without_Partition=pd.DataFrame(result13)


Cases_Per_Ethinicity_ORM_Without_Partition.columns =['Ethinicity','Number_Of_Cases']
Cases_Per_Ethinicity_ORM_Without_Partition
```

	Ethinicity	Number_Of_Cases
0	Non-Hispanic/Latino	1712966
1	Hispanic/Latino	205152
2	Missing	247276
3	NA	455215
4	Unknown	771641



SCENARIO 4:

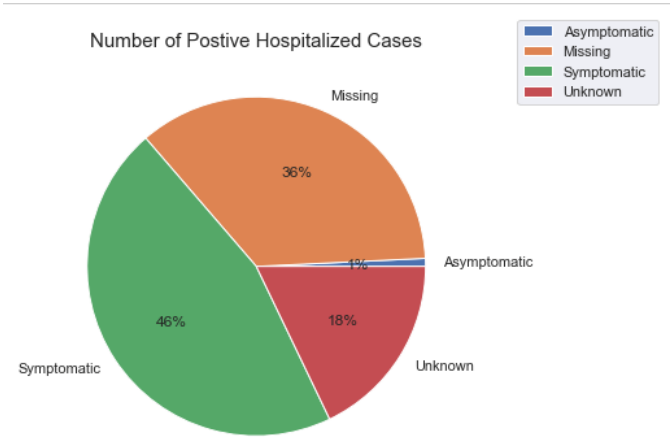
Number of Positive Hospitalized Cases:

```
from sqlalchemy import text
start14=time()
sql14 = text("""select symptom_status,count(*) from
group_project_225.cdc_covid_data_without_partition
group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;""")
result14 = engine.execute(sql14)
end14=time()
tt14=end14-start14
time_taken_orm_without_parition.append(tt14)
Hosp_Positive_Cases_ORM_Without_Partition=pd.DataFrame(result14)

Hosp_Positive_Cases_ORM_Without_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_ORM_Without_Partition
```

SCENARIO 4 : Number of Positive Hospitalized Cases

```
from sqlalchemy import text
start14=time()
sql14 = text("""select symptom_status,count(*) from group_project_225.cdc_covid_data_without_partition
              group by hosp_yn,symptom_status having hosp_yn='Yes' order by hosp_yn,symptom_status;""")
result14 = engine.execute(sql14)
end14=time()
tt14=end14-start14
time_taken_orm_without_parition.append(tt14)
Hosp_Positive_Cases_ORM_Without_Partition=pd.DataFrame(result14)


Hosp_Positive_Cases_ORM_Without_Partition.columns =['Symptom_Status','Number_Of_Cases']
Hosp_Positive_Cases_ORM_Without_Partition
```

	Symptom_Status	Number_Of_Cases
0	Asymptomatic	492
1	Missing	22990
2	Symptomatic	29596
3	Unknown	11544



SCENARIO 5:

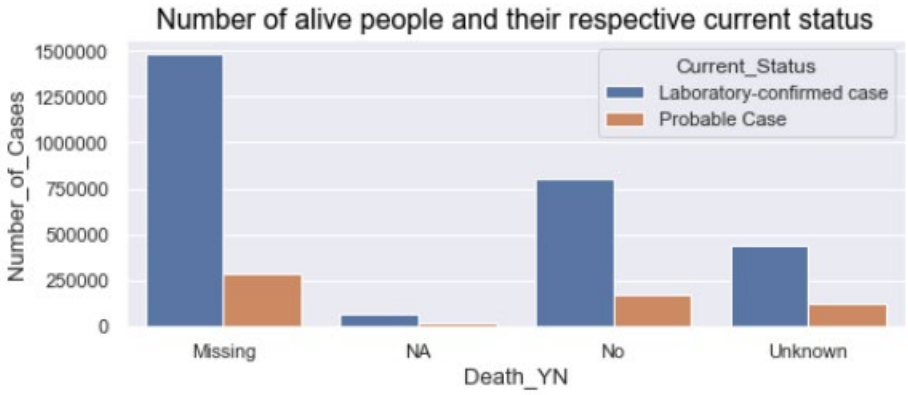
Number of cases and respective current status of alive people:

```
from sqlalchemy import text
start15=time()
sql15 = text("""select death_yn,current_status,count(*) as 'number_people_died' from
group_project_225.cdc_covid_data_without_partition group by death_yn,current_status
having death_yn <> 'Yes' order by death_yn,current_status;""")
result15 = engine.execute(sql15)
end15=time()
tt15=end15-start15
time_taken_orm_without_parition.append(tt15)
Alive_Status_ORM_Without_Partition=pd.DataFrame(result15)
Alive_Status_ORM_Without_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_ORM_Without_Partition
```

SCENARIO 5 : Number of cases and respective current status of alive people

```
from sqlalchemy import text
start15=time()
sql15 = text("""select death_yn,current_status,count(*) as 'number_people_died' from
                group_project_225.cdc_covid_data_without_partition group by death_yn,current_status
                having death_yn <> 'Yes' order by death_yn,current_status;""")
result15 = engine.execute(sql15)
end15=time()
tt15=end15-start15
time_taken_orm_without_parition.append(tt15)
Alive_Status_ORM_Without_Partition=pd.DataFrame(result15)


Alive_Status_ORM_Without_Partition.columns =['Death_YN','Current_Status','Number_of_Cases']
Alive_Status_ORM_Without_Partition
```

	Death_YN	Current_Status	Number_of_Cases
0	Missing	Laboratory-confirmed case	1486202
1	Missing	Probable Case	283238
2	NA	Laboratory-confirmed case	66297
3	NA	Probable Case	19494
4	No	Laboratory-confirmed case	803349
5	No	Probable Case	174811
6	Unknown	Laboratory-confirmed case	437111
7	Unknown	Probable Case	119271



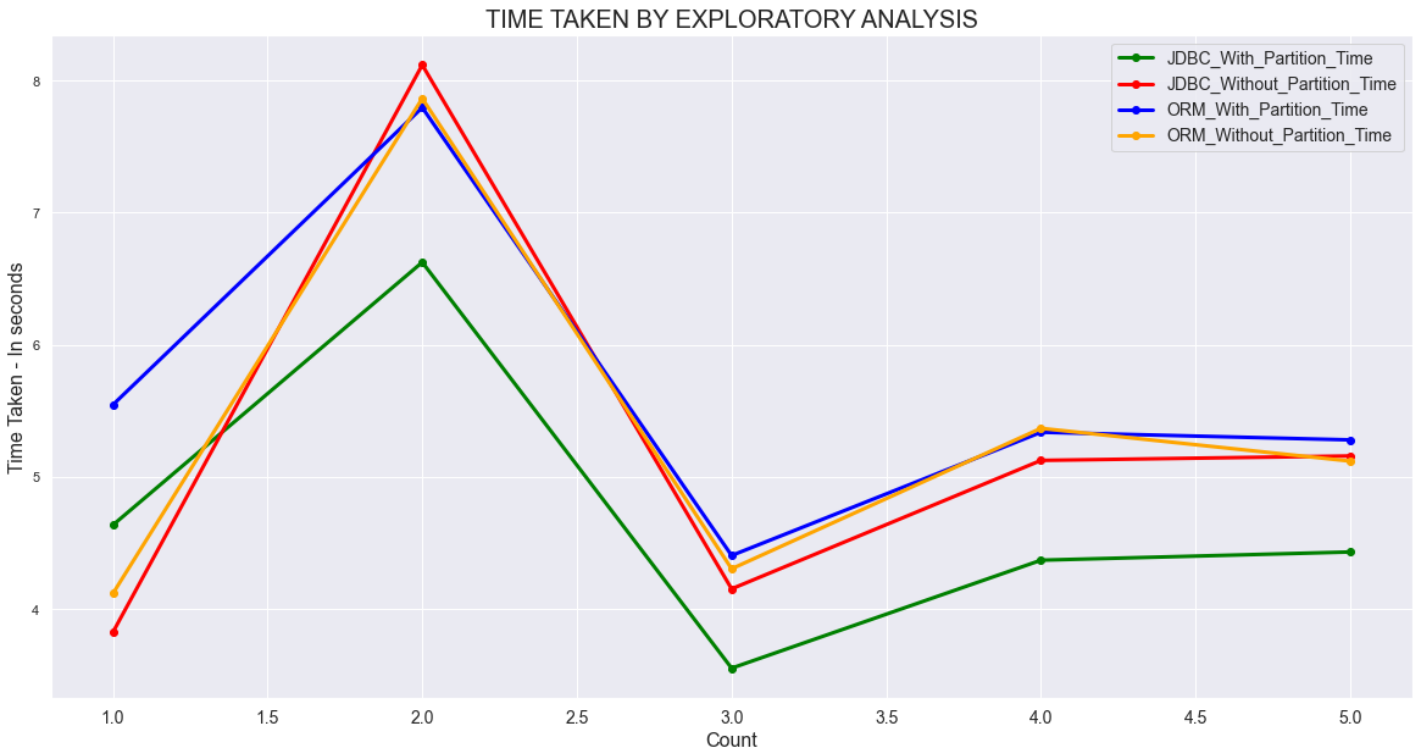
ORM -WITH PARTITION VS WITHOUT PARTITION EXECUTION TIME

	Query_Num	JDBC_With_Partition_Time	JDBC_Without_Partition_Time
0	1	4.634861	3.823150
1	2	6.623851	8.117040
2	3	3.551446	4.151696
3	4	4.368735	5.124085
4	5	4.430888	5.157737

JDBC VS ORM - EXECUTION TIME

	Query_Num	JDBC_With_Partition_Time	JDBC_Without_Partition_Time	ORM_With_Partition_Time	ORM_Without_Partition_Time
0	1	4.634861	3.823150	5.544940	4.116519
1	2	6.623851	8.117040	7.794890	7.863621
2	3	3.551446	4.151696	4.404256	4.303525
3	4	4.368735	5.124085	5.337074	5.366823
4	5	4.430888	5.157737	5.279275	5.118446

TIME TAKEN BY EXPLORATORY ANALYSIS



TIME TAKEN BY EXPLORATORY ANALYSIS – BAR CHART

<End of document>

Submitted by : Gayathri Sundareshwar , Keerthana Gopikrishnan and Deepasha Jenamani

13th May 2022