

EMBEDDED SQL EXECUTION LOGS

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
start = time()
```

IMPORTING PACKAGES

```
In [1]: 1 import mysql.connector
        2 import pandas as pd
        3 import numpy as np
        4 import math
        5 import statistics as stat
        6 import matplotlib.pyplot as plt
        7 import seaborn as sns
        8 import random
        9 import warnings
       10 warnings.filterwarnings('ignore')
       11 %matplotlib inline
       12 from getpass import getpass
       13 from time import time
       14
       15 start = time()
```

GETTING USERNAME & PASSWORD FOR AWS MYSQL DB CONNECTION

```
print(" Connecting to the host 'lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com' ")
global usnm;
usnm=input("Enter username to connect to the AWS MYSQL DB: ")
global pwd;
pwd=getpass("Enter password to connect to the AWS MYSQL DB: ")
```

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

```
1 print(" Connecting to the host 'lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com' ")
2 global usnm;
3 usnm=input("Enter username to connect to the AWS MYSQL DB: ")
4 global pwd;
5 pwd=getpass("Enter password to connect to the AWS MYSQL DB: ")
```

```
Connecting to the host 'lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com'
Enter username to connect to the AWS MYSQL DB: admin
Enter password to connect to the AWS MYSQL DB: .....
```

SQL QUERIES EXECUTION

SCENARIO 1:

Displaying the no of customers present in each country:

```
from mysql.connector import connect, Error
try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm, password=pwd, database="sjsu_movie_db"
    ) as connection:
        sql1 = "select cust_country,count(*) from customer_details group by cust_country order by count(*) desc;"
        with connection.cursor() as cur1:
            cur1.execute(sql1)
            country_wise_customer = pd. DataFrame (cur1. fetchall ())
except Error as e:
    print(e)
country_wise_customer.columns=['Country','Count']
country_wise_customer
```

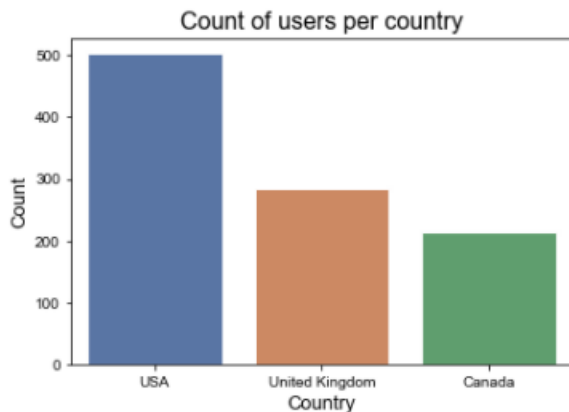
SQL QUERY 1:

Display the No of customers present in each country

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8     ) as connection:
9         sql1 = "select cust_country,count(*) from customer_details group by cust_country order by count(*) desc;"
10        with connection.cursor() as cur1:
11            cur1.execute(sql1)
12            country_wise_customer = pd. DataFrame (cur1. fetchall ())
13except Error as e:
14    print(e)
15
16 country_wise_customer.columns =['Country','Count']
17 country_wise_customer
```

5]:

	Country	Count
0	USA	502
1	United Kingdom	285
2	Canada	214



SCENARIO 2:

Displaying the no of Movies released based on year:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                  user=usnm,
                  password=pwd,
                  database="sjsu_movie_db"
    ) as connection:
        sql2 = "select year,count(*) from Movie_data group by Year order by count(*) desc;;"
        with connection.cursor() as cur2:
            cur2.execute(sql2)
            Year_movie_count = pd. DataFrame (cur2. fetchall ())
except Error as e:
    print(e)

Year_movie_count.columns=['Release Year','Count of Movies Released']
Year_movie_count
```

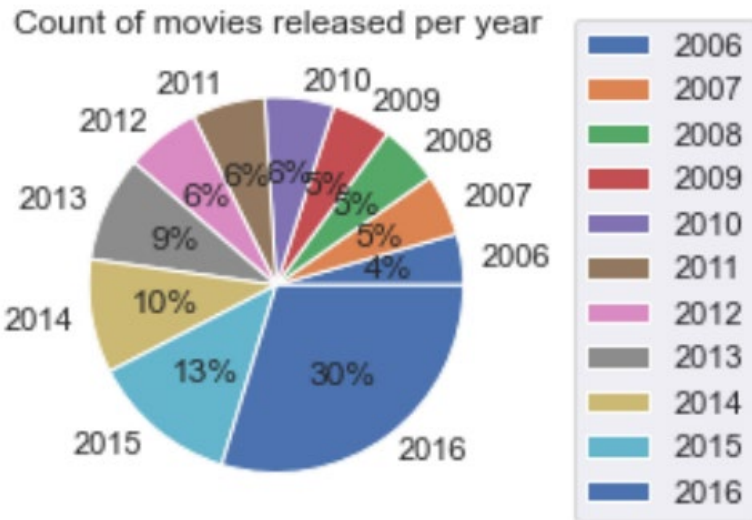
SQL Query 2:

Display the No of Movies Released Based on Year

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                   user=usnm,
6                   password=pwd,
7                   database="sjsu_movie_db"
8     ) as connection:
9         sql2 = "select year,count(*) from Movie_data group by Year order by count(*) desc;;"
10        with connection.cursor() as cur2:
11            cur2.execute(sql2)
12            Year_movie_count = pd. DataFrame (cur2. fetchall ())
13 except Error as e:
14     print(e)
15
16 Year_movie_count.columns=['Release Year','Count of Movies Released']
17 Year_movie_count
```

[7]:

	Release Year	Count of Movies Released
0	2016	297
1	2015	127
2	2014	98
3	2013	91
4	2012	64
5	2011	63
6	2010	60
7	2007	53
8	2008	52
9	2009	51
10	2006	44



SCENARIO 3:

Ranking the top 5 Employees who has responded to the most complaints

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
    ) as connection:
        sql3 = """\
        SELECT c.emp_id, CONCAT(e.emp_first_name, ", ", e.emp_middle_name, ", ", e.emp_last_name) AS 'Employee
        Name',
        COUNT(complaint_id)
        FROM cust_complaints c JOIN employees e USING(emp_id) GROUP BY c.emp_id ORDER BY
        count(complaint_id) DESC LIMIT 5;
        """

        with connection.cursor() as cur3:
            cur3.execute(sql3)
            top_5_employee = pd.DataFrame (cur3.fetchall ())
except Error as e:
    print(e)

top_5_employee.columns = ['Employee ID', 'Employee Name', 'Total complaints resolved']
top_5_employee
```

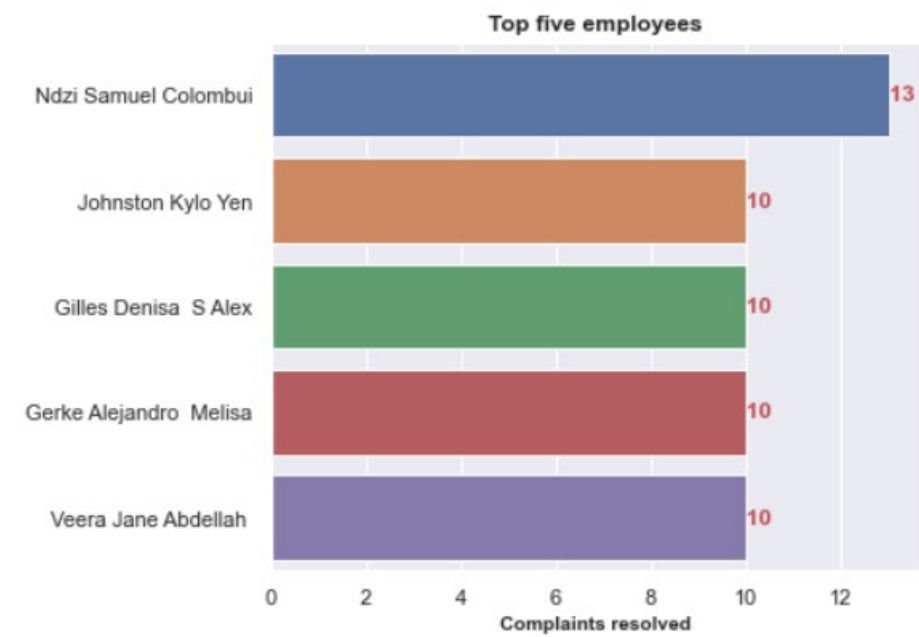
SQL Query 3:

Ranking the Top 5 Employees who has responded to the most complaints

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9                 sql3 = """\
10                    SELECT c.emp_id,CONCAT(e.emp_first_name,',',e.emp_middle_name,',',e.emp_last_name) AS 'Employee Name',
11                           COUNT(complaint_id)
12                    FROM cust_complaints c JOIN employees e USING(emp_id) GROUP BY c.emp_id ORDER BY
13                           count(complaint_id) DESC LIMIT 5;
14                    """
15                 with connection.cursor() as cur3:
16                     cur3.execute(sql3)
17                     top_5_employee = pd.DataFrame (cur3. fetchall ())
18 except Error as e:
19     print(e)
20
21
22 top_5_employee.columns = ['Employee ID','Employee Name','Total complaints resolved']
23 top_5_employee
```

9]:

	Employee ID	Employee Name	Total complaints resolved
0	512211915	Ndzi Samuel Colombui	13
1	521004885	Johnston Kylo Yen	10
2	291255019	Gilles Denisa S Alex	10
3	283386581	Gerke Alejandro Melisa	10
4	615036329	Veera Jane Abdellah	10



SCENARIO 4:

Average time taken to close the tickets:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                  user=usnm,
                  password=pwd,
                  database="sjsu_movie_db"
    ) as connection:
        sql4 = """select
sum(datediff(close_date,complaint_creation_date))/count(datediff(close_date,complaint_creation_date))
as Avg_days_taken
from cust_complaints where resolution_status='Closed'
order by datediff(close_date,complaint_creation_date) desc;"""
        with connection.cursor() as cur4:
            cur4.execute(sql4)
            avg_time=cur4. fetchone ()
            print("Average Time Taken:\t",avg_time[0]," DAYS.")
except Error as e:
    print(e)
```

SQL Query 4:

Average time taken taken to close a ticket.

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                   user=usnm,
6                   password=pwd,
7                   database="sjsu_movie_db"
8     ) as connection:
9         sql4 = """select
10             sum(datediff(close_date,complaint_creation_date))/count(datediff(close_date,complaint_creation_date))
11             as Avg_days_taken
12             from cust_complaints where resolution_status='Closed'
13             order by datediff(close_date,complaint_creation_date) desc;"""
14         with connection.cursor() as cur4:
15             cur4.execute(sql4)
16             avg_time=cur4. fetchone ()
17             print("Average Time Taken:\t",avg_time[0]," DAYS.")
18 except Error as e:
19     print(e)
```

Average Time Taken: 3.3673 DAYS.

SCENARIO 5:

Top 10 most watched movies in USA, UK and CANADA:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
    ) as connection:
        sql5 = """select c.cust_country,m.title,count(u.rank_id) as times_watched from user_watch_history u
        join Movie_data m on u.rank_id=m.rank_id join user_details ud on ud.user_id=u.user_id
        join customer_details c on c.customer_id=ud.customer_id
        group by c.cust_country,m.title order by cust_country,count(rank_id) desc;"""
        with connection.cursor() as cur5:
            cur5.execute(sql5)
            top_10_movies_country_wise = pd.DataFrame (cur5.fetchall ())
except Error as e:
    print(e)
top_10_movies_country_wise.columns=['Country','Title','Times Watched']
top_10_movies_country_wise.Country.unique()
movie_array=top_10_movies_country_wise.to_numpy()
canada_array=[]
UK_array=[]
USA_array=[]
for i in movie_array:
    if i[0]=='Canada':
        canada_array.append(i)
    elif i[0]=='United Kingdom':
        UK_array.append(i)
    elif i[0]=='USA':
        USA_array.append(i)
Canada_df = pd.DataFrame (canada_array, columns= ['Country', 'title','Times Watched'])
UK_df = pd.DataFrame (UK_array, columns= ['Country', 'title','Times Watched'])
USA_df = pd.DataFrame (USA_array, columns= ['Country', 'title','Times Watched'])
sorted_USA = USA_df.sort_values(by='Times Watched', ascending=False)
sorted_UK = UK_df.sort_values(by='Times Watched', ascending=False)
sorted_Canada = Canada_df.sort_values(by='Times Watched', ascending=False)
USA_Top_10=sorted_USA.head(10)
UK_Top_10=sorted_UK.head(10)
Canada_Top_10=sorted_Canada.head(10)
uk_top_10_movies=[]
usa_top_10_movies=[]
canada_top_10_movies=[]
uk_top_10_movies=UK_Top_10['title'].to_numpy()
usa_top_10_movies=USA_Top_10['title'].to_numpy()
canada_top_10_movies=Canada_Top_10['title'].to_numpy()
```

SQL Query 5:

TOP 10 Most watched movies in USA UK and Canada

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql5 = """select c.cust_country,m.title,count(u.rank_id) as times_watched from user_watch_history u
10        join Movie_data m on u.rank_id=m.rank_id join user_details ud on ud.user_id=u.user_id
11        join customer_details c on c.customer_id=ud.customer_id
12        group by c.cust_country,m.title order by cust_country,count(rank_id) desc;"""
13        with connection.cursor() as cur5:
14            cur5.execute(sql5)
15            top_10_movies_country_wise = pd.DataFrame (cur5. fetchall ())
16 except Error as e:
17     print(e)
```

```
1 top_10_movies_country_wise.columns =['Country','Title','Times Watched']
2 top_10_movies_country_wise.Country.unique()
3
```

```
3]: array(['Canada', 'United Kingdom', 'USA'], dtype=object)
```

```
1 movie_array=top_10_movies_country_wise.to_numpy()
2 canada_array=[]
3 UK_array=[]
4 USA_array=[]
5 for i in movie_array:
6     if i[0]=='Canada':
7         canada_array.append(i)
8     elif i[0]=='United Kingdom':
9         UK_array.append(i)
10    elif i[0]=='USA':
11        USA_array.append(i)
12
13 Canada_df = pd.DataFrame (canada_array, columns= ['Country', 'title','Times Watched'])
14 UK_df = pd.DataFrame (UK_array, columns= ['Country', 'title','Times Watched'])
15 USA_df = pd.DataFrame (USA_array, columns= ['Country', 'title','Times Watched'])
16
```

```
1 sorted_USA = USA_df.sort_values(by='Times Watched', ascending=False)
2 sorted_UK = UK_df.sort_values(by='Times Watched', ascending=False)
3 sorted_Canada = Canada_df.sort_values(by='Times Watched', ascending=False)
```

```
1 USA_Top_10=sorted_USA.head(10)
2 UK_Top_10=sorted_UK.head(10)
3 Canada_Top_10=sorted_Canada.head(10)
```

```
1 uk_top_10_movies=[]
2 usa_top_10_movies=[]
3 canada_top_10_movies=[]
4 uk_top_10_movies=UK_Top_10['title'].to_numpy()
5 usa_top_10_movies=USA_Top_10['title'].to_numpy()
6 canada_top_10_movies=Canada_Top_10['title'].to_numpy()
7
```

```
1 top_10_movies=pd.DataFrame()
2 top_10_movies['USA']=usa_top_10_movies
3 top_10_movies['UK']=uk_top_10_movies
4 top_10_movies['Canada']=canada_top_10_movies
5 top_10_movies.head(10)
```


	USA	UK	Canada
0	The Host	Rio	This Beautiful Fantastic
1	The Visit	Total Recall	The Magnificent Seven
2	The Blackcoat's Daughter	Selma	Evil Dead
3	A Million Ways to Die in the West	Warcraft	Sleeping with Other People
4	Blackhat	Friend Request	Carrie
5	Guardians of the Galaxy	The Headhunter's Calling	Bridge of Spies
6	Red Dawn	The Ugly Truth	The Boss
7	Mindhorn	Now You See Me 2	Everest
8	Silver Linings Playbook	American Pastoral	The Skin I Live In
9	Oculus	Pompeii	The Happening

SCENARIO 6:

Display how many screens are used by customers on average:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
    ) as connection:
        sql6 = """select round(sum(no_of_screens)/count(no_of_screens)) as avg_screens from
(select customer_id,count(*) as 'no_of_screens' from user_details group by customer_id
order by customer_id asc) t1;"""
        with connection.cursor() as cur6:
            cur6.execute(sql6)
            avg_screens=cur6.fetchone ()
            print("Average No of Screens used by customers:\t",avg_screens[0]," Screens.")
except Error as e:
    print(e)
```

SQL Query 6:

Display how many screens are used by customers on average

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8     ) as connection:
9         sql6 = """select round(sum(no_of_screens)/count(no_of_screens)) as avg_screens from
10        (select customer_id,count(*) as 'no_of_screens' from user_details group by customer_id
11        order by customer_id asc) t1;"""
12        with connection.cursor() as cur6:
13            cur6.execute(sql6)
14            avg_screens=cur6.fetchone ()
15            print("Average No of Screens used by customers:\t",avg_screens[0]," Screens.")
16 except Error as e:
17     print(e)
```

Average No of Screens used by customers: 3 Screens.

SCENARIO 7:

Total no of complaints recorded grouped by complaint category

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
    ) as connection:
        sql7 = """
        select complaint_category,count(*) as 'Number of Complaints'
        from cust_complaints
        group by complaint_category
        order by count(*) asc;"""
        with connection.cursor() as cur7:
            cur7.execute(sql7)
            count_complaint_category = pd. DataFrame (cur7. fetchall ())
except Error as e:
    print(e)

count_complaint_category.columns =['Complaint Category','No of Complaints recorded']
count_complaint_category
```

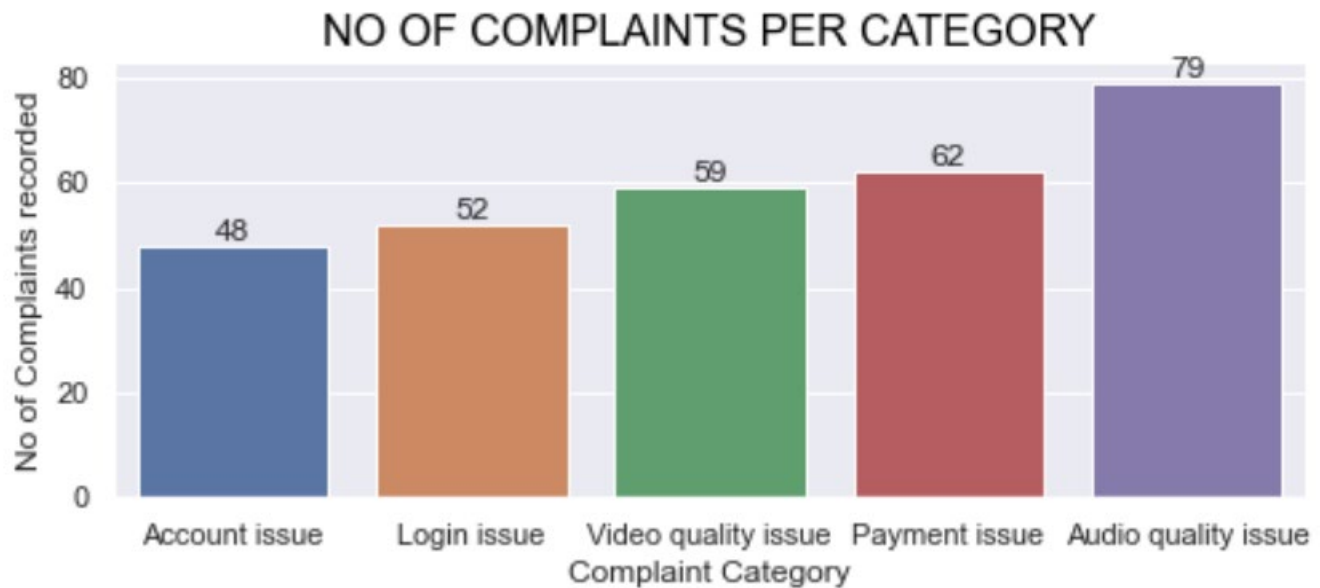
SQL Query 7:

Display the Total No of complaints recorded grouped by Complaint Category

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8     ) as connection:
9         sql7 = """
10        select complaint_category,count(*) as 'Number of Complaints'
11        from cust_complaints
12        group by complaint_category
13        order by count(*) asc;"""
14        with connection.cursor() as cur7:
15            cur7.execute(sql7)
16            count_complaint_category = pd. DataFrame (cur7. fetchall ())
17 except Error as e:
18     print(e)
19
20
21 count_complaint_category.columns =['Complaint Category','No of Complaints recorded']
22 count_complaint_category
```

3]:

	Complaint Category	No of Complaints recorded
0	Account issue	48
1	Login issue	52
2	Video quality issue	59
3	Payment issue	62
4	Audio quality issue	79



SCENARIO 8:

Display total no of complaints closed per month:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
    ) as connection:
        sql8 = """select monthname(close_date) as closed_month,count(*) from cust_complaints
        where resolution_status='Closed' group by month(close_date) order by month(close_date) asc;"""
        with connection.cursor() as cur8:
            cur8.execute(sql8)
            tickets_closed = pd.DataFrame (cur8. fetchall ())
except Error as e:
    print(e)

tickets_closed.columns =['Month','No of Complaints Closed']
tickets_closed
```

SQL Query 8:

Display the Total No of Complaints Closed Per Month

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql8 = """select monthname(close_date) as closed_month,count(*) from cust_complaints
10        where resolution_status='Closed' group by month(close_date) order by month(close_date) asc;"""
11        with connection.cursor() as cur8:
12            cur8.execute(sql8)
13            tickets_closed = pd.DataFrame (cur8. fetchall ())
14 except Error as e:
15     print(e)
16
17 tickets_closed.columns = ['Month','No of Complaints Closed']
18 tickets_closed
```

22]:

	Month	No of Complaints Closed
0	January	61
1	February	86



SCENARIO 9:

Display the total payments made per month:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                  user=usnm,
                  password=pwd,
                  database="sjsu_movie_db"
    ) as connection:
        sql9 = """SELECT DATE_FORMAT(payment_date, '%M, %y') AS payment_month,
sum(total_amount) AS payment_per_month FROM invoice_details
GROUP BY payment_month order by payment_month asc;"""
        with connection.cursor() as cur9:
            cur9.execute(sql9)
            month_amount = pd.DataFrame (cur9. fetchall ())
except Error as e:
    print(e)
month_amount.columns=['Payment Month','Amount']
month_amount['Amount']=pd.to_numeric(month_amount['Amount'])
month_amount
```

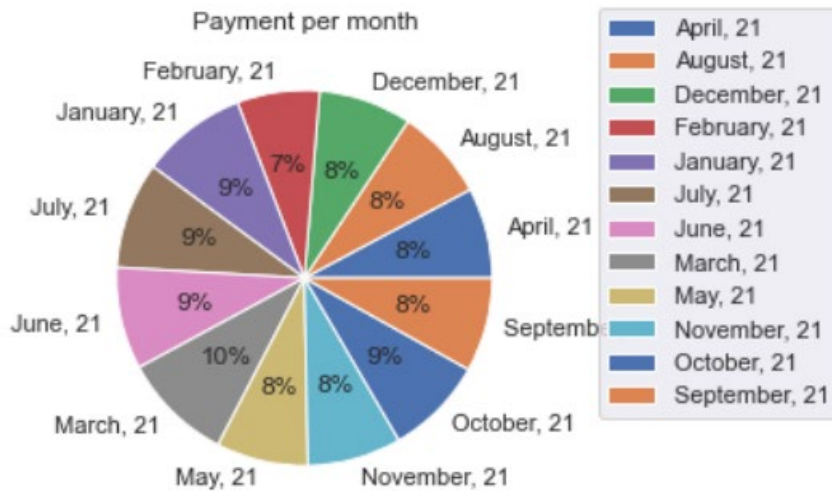
SQL Query 9:

Display the Total Payments Made Per Month

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                   user=usnm,
6                   password=pwd,
7                   database="sjsu_movie_db"
8     ) as connection:
9         sql9 = """SELECT DATE_FORMAT(payment_date, '%M, %y') AS payment_month,
10 sum(total_amount) AS payment_per_month FROM invoice_details
11 GROUP BY payment_month order by payment_month asc;"""
12         with connection.cursor() as cur9:
13             cur9.execute(sql9)
14             month_amount = pd.DataFrame (cur9. fetchall ())
15 except Error as e:
16     print(e)
17
18 month_amount.columns =['Payment Month','Amount']
19 month_amount['Amount']=pd.to_numeric(month_amount['Amount'])
20 month_amount
```

4]:

	Payment Month	Amount
0	April, 21	1556.10
1	August, 21	1576.05
2	December, 21	1596.00
3	February, 21	1416.45
4	January, 21	1815.45
5	July, 21	1835.40
6	June, 21	1755.60
7	March, 21	1895.25
8	May, 21	1576.05
9	November, 21	1615.95
10	October, 21	1695.75
11	September, 21	1615.95



SCENARIO 10:

Top 10 customers based on the no of movies watched:

```
from mysql.connector import connect, Error
```

try:

```
with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
            user=usnm,
            password=pwd,
            database="sjsu_movie_db"
```

) as connection:

```
sql10 = """select concat(c.cust_first_name,' ',c.cust_last_name) as customer_name ,
count(*) as no_of_movies_watched from user_watch_history u
join user_details ud on u.user_id=ud.user_id
join customer_details c on c.customer_id=ud.customer_id
group by customer_name order by count(*) desc limit 10;"""
```

```
with connection.cursor() as cur10:
```

```
cur10.execute(sql10)
```

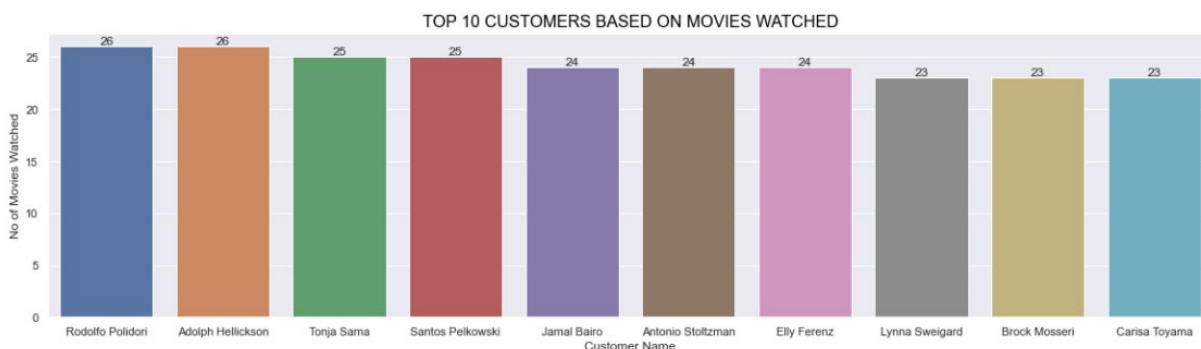
```
top_10_cust = pd. DataFrame (cur10. fetchall ())
```

except Error as e:

```
print(e)
```

```
top_10_cust.columns =['Customer Name','No of Movies Watched']
```

```
top_10_cust
```



SQL Query 10:

Top 10 customers based on the number of movies watched

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql10 = """select concat(c.cust_first_name, ' ', c.cust_last_name) as customer_name ,
10                count(*) as no_of_movies_watched from user_watch_history u
11                join user_details ud on u.user_id=ud.user_id
12                join customer_details c on c.customer_id=ud.customer_id
13                group by customer_name order by count(*) desc limit 10;"""
14        with connection.cursor() as cur10:
15            cur10.execute(sql10)
16            top_10_cust = pd.DataFrame (cur10. fetchall ())
17    except Error as e:
18        print(e)
19
20 top_10_cust.columns = ['Customer Name', 'No of Movies Watched']
21 top_10_cust
```

5]:

	Customer Name	No of Movies Watched
0	Rodolfo Polidori	26
1	Adolph Hellickson	26
2	Tonja Sama	25
3	Santos Pelkowski	25
4	Jamal Bairo	24
5	Antonio Stoltzman	24
6	Elly Ferenz	24
7	Lynna Sweigard	23
8	Brock Mosseri	23
9	Carisa Toyama	23

SCENARIO 11:

Top 5 directors based on user watch history:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
            ) as connection:
                sql11 = """select m.director, count(*) from user_watch_history u join Movie_data m on m.rank_id=u.rank_id
                group by u.rank_id,director order by count(*) desc limit 5; """
                with connection.cursor() as cur11:
                    cur11.execute(sql11)
                    top_5_directors = pd.DataFrame (cur11. fetchall ())
except Error as e:
    print(e)

top_5_directors.columns = ['Director Name', 'Total Movies watched by customers']
top_5_directors
```

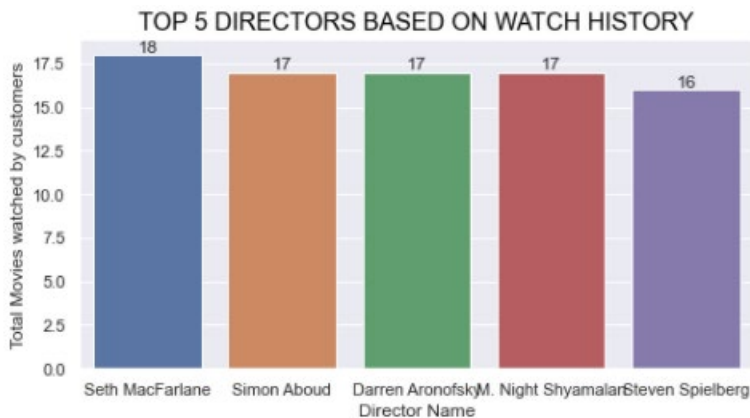
SQL Query 11: ¶

Top 5 directors based on user watch history

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql11 = """select m.director, count(*) from user_watch_history u join Movie_data m on m.rank_id=u.rank_id
10        group by u.rank_id,director order by count(*) desc limit 5; """
11        with connection.cursor() as cur11:
12            cur11.execute(sql11)
13            top_5_directors = pd.DataFrame (cur11. fetchall ())
14    except Error as e:
15        print(e)
16
17 top_5_directors.columns =['Director Name','Total Movies watched by customers']
18 top_5_directors
```

27]:

	Director Name	Total Movies watched by customers
0	Seth MacFarlane	18
1	Simon Aboud	17
2	Darren Aronofsky	17
3	M. Night Shyamalan	17
4	Steven Spielberg	16



SCENARIO 12:

Total no of complaints based on severity:

```
from mysql.connector import connect, Error
try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
            ) as connection:
        sql12 = "Select severity,count(*) as 'number of complaints' from cust_complaints group by severity; "
        with connection.cursor() as cur12:
            cur12.execute(sql12)
            count_severity = pd.DataFrame (cur12. fetchall ())
except Error as e:
    print(e)
count_severity.columns =['Severity level','No of complaints']
count_severity
```

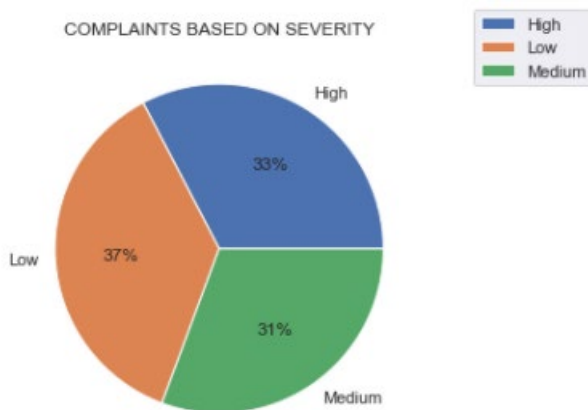

SQL Query 12:

Total no of Complaints based on severity

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql12 = "Select severity,count(*) as 'number of complaints' from cust_complaints group by severity; "
10        with connection.cursor() as cur12:
11            cur12.execute(sql12)
12            count_severity = pd.DataFrame (cur12. fetchall ())
13 except Error as e:
14     print(e)
15
16 count_severity.columns = ['Severity level','No of complaints']
17 count_severity
```

:9]:

	Severity level	No of complaints
0	Medium	92
1	Low	110
2	High	98



SCENARIO 13:

Display the average run time of movies:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
            ) as connection:
        sql13 = "select avg(runtime) from Movie_data ;"
        with connection.cursor() as cur13:
            cur13.execute(sql13)
            avg_runtime=cur13. fetchone ()
            print("Average runtime of the movies is:\t",avg_runtime[0]," mins.")
except Error as e:
    print(e)
```

SQL Query 13:

Display the Average runtime of movies

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql13 = "select avg(runtime) from Movie_data ;"
10        with connection.cursor() as cur13:
11            cur13.execute(sql13)
12            avg_runtime=cur13.fetchone ()
13            print("Average runtime of the movies is:\t",avg_runtime[0]," mins.")
14 except Error as e:
15     print(e)
```

Average runtime of the movies is: 113.1720 mins.

SCENARIO 14:

Display the total no of complaints registered per month:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
            ) as connection:
        sql14 = """Select date_format((complaint_creation_date),'%M'),count(*) as 'No of Complaints per Month'
        from cust_complaints group by month(complaint_creation_date);"""
        with connection.cursor() as cur14:
            cur14.execute(sql14)
            complaints_per_month = pd.DataFrame (cur14.fetchall ())
except Error as e:
    print(e)

complaints_per_month.columns=['Month','No of complaints']
complaints_per_month
```

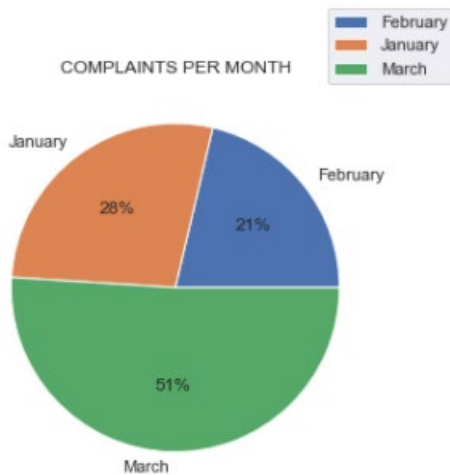
SQL Query 14:

Display the Total No of complaints registered per month

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql14 = """Select date_format((complaint_creation_date), '%M'), count(*) as 'No of Complaints per Month'
10        from cust_complaints group by month(complaint_creation_date);"""
11        with connection.cursor() as cur14:
12            cur14.execute(sql14)
13            complaints_per_month = pd.DataFrame (cur14. fetchall ())
14 except Error as e:
15     print(e)
16
17 complaints_per_month.columns = ['Month', 'No of complaints']
18 complaints_per_month
```

2]:

	Month	No of complaints
0	January	83
1	February	64
2	March	153



SCENARIO 15:

Display the average salary of employees based on the designation:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                  user=usnm,
                  password=pwd,
                  database="sjsu_movie_db"
    ) as connection:
        sql15 = """\
        SELECT emp_position, AVG(emp_salary)
        FROM employees
        GROUP BY emp_position;
        """

        with connection.cursor() as cur15:
            cur15.execute(sql15)
            avg_salary = pd.DataFrame (cur15. fetchall ())
except Error as e:
    print(e)

avg_salary.columns = ['Designation', 'Average Salary']
avg_salary
```

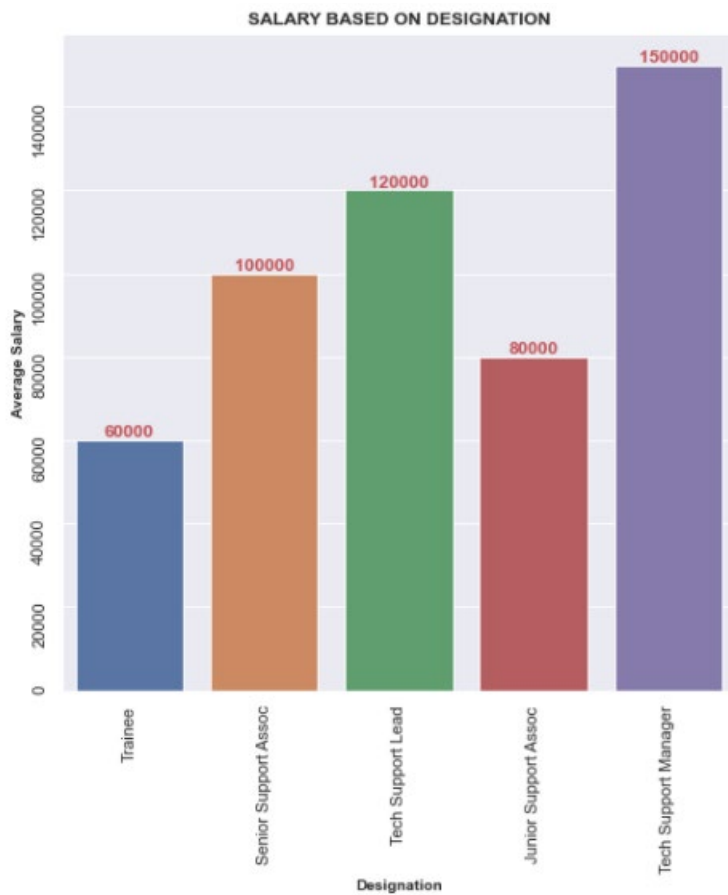
SQL Query 15:

Display the average salary of an Employee based on the designation

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                   user=usnm,
6                   password=pwd,
7                   database="sjsu_movie_db"
8     ) as connection:
9         sql15 = """\
10         SELECT emp_position, AVG(emp_salary)
11         FROM employees
12         GROUP BY emp_position;
13         """
14         with connection.cursor() as cur15:
15             cur15.execute(sql15)
16             avg_salary = pd.DataFrame (cur15. fetchall ())
17 except Error as e:
18     print(e)
19
20 avg_salary.columns = ['Designation', 'Average Salary']
21 avg_salary
```

4]:

	Designation	Average Salary
0	Trainee	60000.0000
1	Senior Support Assoc	100000.0000
2	Tech Support Lead	120000.0000
3	Junior Support Assoc	80000.0000
4	Tech Support Manager	150000.0000



SCENARIO 16:

Display the total no of employees per designation based on employment status:

```
from mysql.connector import connect, Error

try:
    with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
                user=usnm,
                password=pwd,
                database="sjsu_movie_db"
    ) as connection:
        sql16 = """Select employment_status, emp_position, count(*) as 'Number of Employees'
        from employees group by emp_position,employment_status order by employment_status;"""
        with connection.cursor() as cur16:
            cur16.execute(sql16)
            no_of_emp = pd.DataFrame (cur16. fetchall ())
except Error as e:
    print(e)

no_of_emp.columns =['Employment Status','Designation','No of Employees']
no_of_emp
```

SQL Query 16:

Display the Total No of employees per designation based on employment status

```
1 from mysql.connector import connect, Error
2
3 try:
4     with connect(host="lab-assignment-225.cibzfcia066j.us-east-1.rds.amazonaws.com",
5                 user=usnm,
6                 password=pwd,
7                 database="sjsu_movie_db"
8             ) as connection:
9         sql16 = """Select employment_status, emp_position, count(*) as 'Number of Employees'
10                from employees group by emp_position,employment_status order by employment_status;"""
11         with connection.cursor() as cur16:
12             cur16.execute(sql16)
13             no_of_emp = pd.DataFrame (cur16. fetchall ())
14 except Error as e:
15     print(e)
16
17 no_of_emp.columns = ['Employment Status','Designation','No of Employees']
18 no_of_emp
```

6]:

	Employment Status	Designation	No of Employees
0	Permanent Worker	Trainee	15
1	Permanent Worker	Senior Support Assoc	4
2	Permanent Worker	Junior Support Assoc	6
3	Permanent Worker	Tech Support Lead	3
4	Permanent Worker	Tech Support Manager	4
5	Temporary Worker	Trainee	5
6	Temporary Worker	Tech Support Lead	4
7	Temporary Worker	Senior Support Assoc	5
8	Temporary Worker	Tech Support Manager	2
9	Temporary Worker	Junior Support Assoc	2