Data Analysis Using Python

```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")

#do not change the predefined function names

#Task 1: Remove columns that are not needed in our analysis.
# Remove Url_spotify, Uri, Key, Url_youtube, Description
def Remove_columns():
    #do not remove following line of code
    df = pd.read_csv('Spotify_Youtuben.csv')

    #WRITE YOUR CODE HERE
    df =
df.drop(columns=["Url_spotify","Uri","Key","Url_youtube","Description"
])
    #return dataframe
    return df


#Task 2: Check for the null values
def no_of_null_values():
    #Do not remove the following code statment
    df=Remove_columns()
    df= df.isnull().sum()
    #WRITE YOUR CODE HERE TO CHECK THE NO OF NULL VALUES AND RETURNS
THE SAME

    #return sum of null values by columns
    return df


#Task 3: Handle the null values replace int value with 0 and other
values with NA
def Handle_Null_values():
    #Do not remove the following code statment
    df=Remove_columns()

    df=df.fillna(0)
    df=df.dropna(subset=['Title','Channel'])
    df=df.replace('/','',regex=True)

    #df.select_dtypes(include = 'int').fillna(0, inplace = True)
    #df.fillna("NA",inplace = True)
    #WRITE YOUR CODE HERE ACCORDING TO THE DESCRIPTION
```

```
    #return dataframe
    return df

#Task 4: CHECK FOR DUPLICATES AND REMOVE THEM KEEPING THE FIRST VALUE
def drop_the_duplicates():
    #Do not remove the following code statment
    df=Handle_Null_values()
    df.drop_duplicates(keep='first', inplace=True)
    #WRITE YOUR CODE HERE
    #return dataframe
    return df
    #drop_the_duplicates()

#Task 5: CONVERT millisecond duration to minute for a better
understanding
def convert_milisecond_to_Minute():
    #Do not remove the following code statment
    df=drop_the_duplicates()

    df['Duration_ms']=
pd.to_numeric(df['Duration_ms'],errors='coerce')
    df['Duration_ms'] = df['Duration_ms'] / 60000
    #return dataframe
    return df

#Task 6: Rename the modified column to Duration_min
def rename_modified_column():
    #Do not remove the following code statment
    df=convert_milisecond_to_Minute()
    df.rename(columns={'Duration_ms':'Duration_min'},inplace=True)
    #WRITE YOUR CODE HERE

    #return dataframe
    return df

#Task 7: Remove irrelevant 'Track' name that starts with ?
def Irrelevant_Track_name():
    #Do not remove the following code statment
    df=rename_modified_column()

    #df.drop(df[df['Track'].str.startswith('?')].index, inplace=False)
    #df = df[df['Track'].str.contains('^[^?]')]
    #df = df.drop[df['Track'].str.startswith('^[^?]')]
    df = df[~df['Track'].str.startswith('?')|df['Track'].isnull()]
    #WRITE YOUR CODE HERE

    #return dataframe
    return df
```

```python
#Task 8: Calculate the Energy to Liveness ratio for each track and
store it in columns 'EnergyLiveness'
def Energy_to_liveness_Ratio():
    #Do not remove the following code statment
    df=Irrelevant_Track_name()

    df['EnergyLiveness'] = df['Energy'] / df['Liveness']
    #WRITE YOUR CODE HERE
    #return dataframe
    return df

#Task 9: change the datatype of 'views' to float for further use
def change_the_datatype():
    #Do not remove the following code statment
    df=Energy_to_liveness_Ratio()
    df['Views'] = df['Views'].astype(float)
    #WRITE YOUR CODE HERE

    #return dataframe
    return df

#Task 10: compare the views and stream columns to infer
# that the song track was more played on which platform, youtube or
Spotify.
# Create a column named most_playedon which will have two values.
# Spotify and Youtube,If a song track is most played on youtube then
# the most_played on column will have youtube as the value for that
particular song
def compare_the_views():
    #Do not remove the following code statment
    df=change_the_datatype()
    df['Stream'] = df['Stream'].astype(float)

    df['most_playedon'] = df.apply(lambda x: 'Spotify' if x['Stream']
> x['Views'] else 'Youtube', axis=1)
    df['most_playedon']=df['most_playedon'].str.title()
    #WRITE YOUR CODE HERE

    return df


#Task 11: export the cleaned dataset to CSV to "cleaned_dataset.csv"
def export_the_cleaned_dataset():
    #Do not remove the following code statment
    df=compare_the_views()
    df.to_csv("cleaned_dataset.csv",index=False)

    #WRITE YOUR CODE HERE
    #create csv file "cleaned_dataset.csv" using dataframe
```

```
#TASK 12
#follow the instruction in the Task 13 description and complete the
task as per it.

#check if mysql table is created using "cleaned_dataset.csv"
#Use this final dataset and upload it on the provided database for
performing analysis in  MySQL
#To run this task click on the terminal and click on the run projec
```

Business Problem solved using SQL

Which is the most viewed song track on youtube?

SELECT Track, Views FROM cleaned_dataset ORDER BY Views DESC LIMIT 1;

Total Records Fetched: 1 You will see maximum 50 records in your result Headers: Track, Views,

```
Values:
Despacito, 8079649362,
```

Which Song track is streamed most on Spotify?

```
SELECT Track, Stream
FROM cleaned_dataset
ORDER BY Stream DESC
LIMIT 1
```

Total Records Fetched: 1 You will see maximum 50 records in your result Headers: Track, Stream,

```
Values:
Blinding Lights, 3386520288,
```

EnergyLiveness ratio is one of the popular ways to measure the quality of the song, which are the top 5 songs that have the highest energyliveness ratio

```
SELECT Track, (EnergyLiveness) AS EnergyLivenessRatio
FROM cleaned_dataset
ORDER BY EnergyLivenessRatio DESC
LIMIT 5
```

***Your Output: Total Records Fetched: 5 You will see maximum 50 records in your result Headers: Track, EnergyLivenessRatio,

```
Values:
These Words, nan,
Rain in the Early Morning, nan,
```

```
Dakota, 9.989258861,
2 Baddies, 9.989154013,
Over The Hills And Far Away, 9.989082969
```

let us assume a situation where an artist named Black Eyed Peas wants to analyze his songs. The artist wants to know which platform is capable of keeping his song track more engaged. To check this he assigns you this task and wants you to report to him where his song tracks are more played on. compare the platforms.

```
SELECT COUNT(Track) AS TotalTrack, most_playedon
FROM cleaned_dataset
WHERE Artist = "Black Eyed Peas"
GROUP BY most_playedon
ORDER BY TotalTrack DESC;
```

Gorillaz wants to know their most liked song on youtube. Report to them with their most liked song along with the Energy and Tempo of the song.

```
SELECT Track, Energy, Tempo, Likes
FROM cleaned_dataset
WHERE Artist = 'Gorillaz'
ORDER BY Likes DESC
LIMIT 1;
```

***Your Output: Total Records Fetched: 1 You will see maximum 50 records in your result Headers: Track, Energy, Tempo, Likes,

```
Values:
Feel Good Inc., 0.705000, 138.559, 6220896,
```

Which Album types are more prominent on Spotify?

```
SELECT
Album_type, count(Album_type)
from cleaned_dataset
GROUP BY Album_type ORDER BY count(Album_type) DESC;
```

***Your Output: Total Records Fetched: 3 You will see maximum 50 records in your result Headers: Album_type, count(Album_type),

```
Values:
album, 14834,
single, 4973,
compilation, 787,
```

Spotify's most loved song tracks are to be declared soon. Help Spotify choose the top 5 most streamed+youtube viewed song track

```
SELECT Track, (Stream + Views) AS Total
FROM cleaned_dataset
GROUP BY Track
```

```
ORDER BY Total DESC
LIMIT 5;
```