Question 1: What is Pandas, and why is it commonly used in data cleaning tasks?

Ans: Pandas is open source python library that is used for data cleaning, preprocessing, transformation, analysis and manipulation.
It have 2 data structures series and Dataframe.
Series is 1 Dimensional Data structure, Dataframe can be 2 Dimensional or Multi-Dimensional data structure.
It is used to handling missing values, removing duplicates, filtering, sorting, filling in missing data removing outliers etc. Hence used in cleaning the data
It gives quick statistical and descriptive analysis of Data.
We can merge 2 datasets using pandas.
Also it can be used with other libraries like Seaborn, numpy, matplotlib etc.

Question 2: Given a DataFrame with missing values, how would you check for missing values in each column and count the total number of missing values?

Ans: We can find the missing values of all the columns in a dataframe by a single code, consider df as a Dataframe
df.isnull().sum()
To get all the total of missing values in d Dataframe from all the columns we can use following code
df.isnull().sum().sum()

Question 3: How can you remove duplicates from a DataFrame while retaining the first occurrence of each unique row?

Ans: Duplicate entries affect Data Analysis hence we have to drop it
We can find duplicate entries in column by using:
df.duplicated()
And we can drop duplicate entries while retaining first occurrence by using:
df.drop_duplicates()

Question 4: If you have a DataFrame with a column containing string values, how can you convert all the values in that column to lowercase?

Ans: Suppose Column name is- Sentimental_grcry
So to convert all string values in lowercase we can use
df.Sentiment_grcry.str.lower()

Question 5: How do you replace missing values in a DataFrame with a specific value, like 0, for a particular column?

Ans: We can fill any value using fillna function
df. Sentimental_grcry.fillna(0) – To fill 0
df. Sentimental_grcry.fillna(df.Sentimental_grcry.mean())- To fill mean value of same column where there is a normal distribution

Question 6: If you have a DataFrame with a datetime column, how can you extract the year, month, and day into separate columns?

Ans: We can extract, month year date hour day minute, weekday, etc using date time module

Consider review_date is the column
df[‘Year’] =df[‘review_date].dt.year
df[‘Month’] =df[‘review_date].dt.month
df[‘Date] =df[‘review_date].dt.day

Question 7: How can you filter rows in a DataFrame where a specific column's values meet a certain condition (e.g., all rows where 'age' is greater than 30)?

Ans: Here we are filtering the data where, age is less thab 30 and they are from Nagpur.
We can use filter condition df[(df[‘Age’]<30 & (df[‘City’]==(‘Nagpur’)]

Question 8: What is the purpose of the .apply() function in Pandas, and how would you use it to create a new column based on values from existing columns?

Ans: In Pandas, the apply() function allows for the application of a function across an axis of a DataFrame or a specific axis of a Series.
You can use a custom or built-in function to apply a custom or built-in function to every element or row/column in the DataFrame.
Here we are applying apply function using def function and from column product title we are removing quotation mark

```
def remove_quotation_marks(org):
    org=str(org)
    org=org.split("")
    org=org[1]
    return org
df['product_title_new']=df['product_title'].apply(remove_quotation_marks) ,axis=1
```

Question 9: Suppose you want to merge two DataFrames, 'df1' and 'df2,' on a common column 'key.' How would you perform this merge operation in Pandas?

Ans: df_new= pd.merge(df1, df2, on=’key’, how=’inner’)

Question 10: You have a DataFrame with a column containing messy text data. How can you clean and standardize the text data (e.g., remove punctuation and convert to lowercase) in that column

Ans: Here we are using data=df, column= review_title

```
def remove_punctuation_marks(org):
    org=str(org)
    org=org.split("")
    org=org[1]
    return org
df['review_title']=df['review_title'].apply(remove_punctuation_marks)
```

to convert it to lower case

```
df.review_title.str.lower()
```