Question 1

```
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark.sql.types import *

spark = SparkSession.builder.appName("Emp").getOrCreate()

emp = spark.read.csv("emp_data.csv",header=True,inferSchema=True)
emp.show()

missing_counts = emp.select([count(when(col(c).isNull() | isnan(col(c)) ,c )).alias(c) for c in emp.columns])
missing_counts.show()

emps = emp.fillna({"LastName" : "Unknown"})
emps.show()

emp = emp.dropna(subset = ["EmpID","StartDate"])
emp.show()

emp = emp.withColumn("Current Employee Rating",
                when(col("Current Employee Rating") < 1 , 1)
                .when(col("Current Employee Rating") > 5 , 5)
                .otherwise("Current Employee Rating"))
emp.show()

emp.select("LocationCode").distinct().show()

df = emp.dropDuplicates()
df.show()

df = df.groupBy("DepartmentType", "Title").count().orderBy("DepartmentType", "Title")
df.show()
```

```
df = emp.withColumn(
   "Performance Score",
   when(col("Performance Score") == "Fully Meets", 1)
   .when(col("Performance Score") == "Exceeds", 2)
   .when(col("Performance Score") == "Meets", 3)
   .otherwise(0)
)
highest_pref = (
    df.orderBy("DepartmentType", col("Performance Score").desc())
    .dropDuplicates(["DepartmentType"])
    .select("DepartmentType","EmpID","Firstname","LastName", "Performance Score")
)
highest_pref.show()
```

Question 2

```
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark.sql.types import *

spark = SparkSession.builder.appName("Sales").getOrCreate()

df = spark.read.csv("Sales Data.csv",header=True,inferSchema=True)
df.show()
df.printSchema()

df_null = df.select([count(when(col(c).isNull() | isnan(col(c)) , c)).alias(c) for c in df.columns])
df_null.show()

numerical_columns = ['Sales', 'Quantity Ordered']
for col_name in numerical_columns:
   mean_value = df.select(mean(col_name)).collect()[0][0]
   df = df.fillna({col_name: mean_value})
```

```
# Drop rows with null values in critical columns
df = df.dropna()
df.show()

df = df.dropDuplicates()
df.show()

df = df.withColumn("Sales" , col("Sales").cast("float"))
df = df.withColumn("Quantity Ordered" , col("Quantity Ordered").cast("integer"))
df = df.withColumn("Price Each" , col("Price Each").cast("float"))
df.printSchema()

columns_to_check = ["Sales","Quantity Ordered","Price Each"]
for c in columns_to_check:
    df = df.filter(col(c)>=0)
    df.show()

df.groupBy("Product").sum("Sales").withColumnRenamed("sum(Sales)" , "Total_Sales").show()
```

Question 3

```
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark.sql.types import *

spark = SparkSession.builder.appName("jobs").getOrCreate()

jobs = spark.read.csv("Cleaned_DS_Jobs.csv", header=True, inferSchema=True)
jobs.show()

jobs = jobs.withColumn("min_salary",regexp_extract(col("Salary Estimate"),r"(\d+)-(\d+)",1).cast("float"))
jobs = jobs.withColumn("max_salary",regexp_extract(col("Salary Estimate"),r"(\d+)-(\d+)",2).cast("float"))
jobs.show()
```

```
jobs = jobs.withColumn("avg_salary",(col("min_salary")+col("max_salary"))/2)
jobs.show()

jobs = jobs.withColumn("Rating" , when((col("Rating")==0) | (col("Rating")==-1),1).otherwise(col("Rating")))
jobs.show()

jobs = jobs.fillna(-1)

jobs_title = jobs.groupBy("Job Title").agg(mean("avg_salary")).alias("avg_salary")
jobs_title.show()

job_size = jobs.groupBy("Size").agg(mean("avg_salary").alias("avg_salary"))
job_size.show()

test="test.csv"
jobs.write.csv("test",header=True,mode="overwrite")
```