from pyspark.sql import SparkSession  
from pyspark.sql.functions import col, avg, coalesce  
from pyspark.sql.window import Window  
  
  
spark = SparkSession.builder.appName("Employee Performance Review Analysis").getOrCreate()  
  
  
employee=[  
 (1, '2024-01-10', 'Engineering', 5, 'John'),  
 (2, '2024-01-11', 'HR', 4, 'Jane'),  
 (3, '2024-01-12', 'Sales', 3, 'Sam'),  
 (4, '2024-02-01', 'Engineering', 5, 'John'),  
 (1, '2024-03-10', 'Engineering', 4, 'Jane'),  
 (2, '2024-03-11', 'HR', None, 'Sam')  
]  
# Create DataFrame  
columns = ["emp\_id", "review\_date", "department", "rating", "reviewer"]  
df = spark.createDataFrame(employee, schema=columns)  
df.show()  
  
#filter  
#df\_filtered = df.filter(df.rating >= 4)  
#df\_filtered.show()  
  
#Handle Null  
#window\_spec = Window.partitionBy('department')  
#df\_filled = df.withColumn('rating', coalesce(df.rating, avg('rating').over(window\_spec)))  
#df\_filled.show()  
  
#drop duplicate  
#Remove duplicate reviews by 'emp\_id' and 'review\_date'.  
#df\_no\_duplicates = df.dropDuplicates(['emp\_id', 'review\_date'])  
#df\_no\_duplicates.show()  
  
#Select Specific Columns  
#Select 'emp\_id', 'department', and 'rating' columns.  
  
#df\_selected = df.select('emp\_id', 'department', 'rating')  
#df\_selected.show()  
  
#Grouping and Aggregating  
#Calculate the average rating per department.  
  
#df\_grouped = df.groupBy('department').agg({'rating': 'avg'})  
#df\_grouped.show()  
  
'''  
df\_employees=[  
 (1,"Amit"),  
 (2,"Rajesh"),  
 (3,"Mukesh"),  
 (4,"Rekha")  
]  
columns2=["emp\_id","employee\_name"]  
df2=spark.createDataFrame(data = df\_employees, schema = columns2)  
  
# Assuming df\_employees is another DataFrame that contains employee details  
df\_joined = df.join(df2, on='emp\_id', how='inner')  
df\_joined.show()  
'''  
#UNION  
'''  
Union of DataFrames  
Union with another 'df\_new\_reviews' DataFrame containing additional reviews.  
  
df\_new\_reviews=[  
 (1, '2024-01-10', 'QA', 5, 'harry'),  
 (5, '2024-05-11', 'HR', 4, 'sane'),  
  
]  
columns2 = ["emp\_id", "review\_date", "department", "rating", "reviewer"]  
df2 = spark.createDataFrame(df\_new\_reviews, schema=columns)  
df\_union = df.union(df2)  
df\_union.show()  
'''  
  
#View and SQL  
'''  
df.createOrReplaceTempView('performance\_reviews')  
sql\_result = spark.sql('SELECT emp\_id, AVG(rating) as avg\_rating FROM performance\_reviews GROUP BY emp\_id')  
sql\_result.show()  
'''  
  
#Window Functions  
#Calculate the cumulative average rating for each employee over time.  
  
window\_spec = Window.partitionBy('emp\_id').orderBy('review\_date')  
df\_with\_cumulative\_avg = df.withColumn('cumulative\_avg', avg('rating').over(window\_spec))  
df\_with\_cumulative\_avg.show()