Python Summer Party Challenge

by Interview Master

Day 1 of 15

WhatsApp

You are a Product Analyst on the WhatsApp team investigating group messaging dynamics. Your team wants to understand how large groups are being used and their messaging patterns. You'll leverage data to uncover insights about group participation and communication behaviors.

Challenge Questions

Q1:

What is the maximum number of participants among WhatsApp groups that were created in October 2024? This metric will help us understand the largest group size available.

Q2:

What is the average number of participants in WhatsApp groups that were created in October 2024? This number will indicate the typical group size and inform our group messaging feature considerations.

Q3:

For WhatsApp groups with more than 50 participants that were created in October 2024, what is the average number of messages sent? This insight will help assess engagement in larger groups and support recommendations for group messaging features.

Want to try this yourself?

Join the Challenge

Sign up for the Python Summer Party Challenge and solve 21 days of data science problems

www.interviewmaster.ai/python-party

Or keep scrolling to see my solutions

My Solution - Q1

Day 1 Python Challenge

```
# Note: pandas and numpy are already imported as pd and np
# The following tables are loaded as pandas DataFrames with th
e same names: dim_groups
# Please print your final result or dataframe

# Step 1: Convert 'created_date' to datetime if it's not alrea
dy
dim_groups['created_date'] = pd.to_datetime(dim_groups['create
d_date'])

# Step 2: Filter for groups created in October 2024
october_2024_groups = dim_groups[
    (dim_groups['created_date'].dt.year == 2024) &
    (dim_groups['created_date'].dt.month == 10)
]

# Step 3: Find the maximum participant count
max_participants = october_2024_groups['participant_count'].ma
x()

# Step 4: Present results as a DataFrame
result_df = pd.DataFrame({
    'Max Participants in Oct 2024 Groups': [max_participants]
})

# Step 5: Display the result
print(result_df)
```

InterviewMaster.AI

My Solution - Q2

Day 1 Python Challenge

```
# Step 1: Convert 'created_date' to datetime if needed
dim_groups['created_date'] = pd.to_datetime(dim_groups['create
d_date'])

# Step 2: Filter for groups created in October 2024
october_groups = dim_groups[
    (dim_groups['created_date'].dt.year == 2024) &
    (dim_groups['created_date'].dt.month == 10)
]

# Step 3: Calculate average participant count
average_participants = october_groups['participant_count'].mea
n()

# Step 4: Present the result as a DataFrame
result_df = pd.DataFrame({
    'Average Participants in Oct 2024 Groups': [average_partic
ipants]
})

# Step 5: Display the result
print(result_df)
```

InterviewMaster.AI

My Solution - Q3

Day 1 Python Challenge

```
# Step 1: Ensure datetime format
dim_groups['created_date'] = pd.to_datetime(dim_groups['create
d_date'])

# Step 2: Filter for October 2024 groups
october_groups = dim_groups[
    (dim_groups['created_date'].dt.year == 2024) &
    (dim_groups['created_date'].dt.month == 10)
]

# Step 3: Filter for groups with more than 50 participants
large_october_groups = october_groups[october_groups['particip
ant_count'] > 50]

# Step 4: Calculate average number of messages
average_messages = large_october_groups['total_messages'].mean
()

# Step 5: Present result
result_df = pd.DataFrame({
    'Avg Messages in Oct 2024 Large Groups (>50 participant
s)': [average_messages]
})

# Step 6: Display
print(result_df)
```

Ready for your own challenge?

Try this yourself by signing up for the Python Summer Party challenge:

www.interviewmaster.ai/python-party

InterviewMaster.AI