Double-click (or enter) to edit

step 1:

```
import pandas as pd

# Sample data based on the provided image
data = {
    "student_id": [
        101, 101, 101, 101, 101, 101, 102, 102, 102, 102,
        103, 103, 103, 103, 103, 103, 103, 103, 103, 104, 104, 104
    ],
    "attendance_date": [
        "2024-01-01", "2024-01-02", "2024-01-03", "2024-01-04", "2024-01-05", "2024-01-06",
        "2024-02-01", "2024-02-02", "2024-02-03", "2024-02-04",
        "2024-03-01", "2024-03-02", "2024-03-03", "2024-03-04", "2024-03-05", "2024-03-06", "2024-
        "2024-04-01", "2024-04-02", "2024-04-03"
    ],
    "status": [
        "Absent", "Absent", "Absent", "Absent", "Absent", "Present",
        "Absent", "Absent", "Absent", "Present",
        "Absent", "Absent", "Absent", "Absent", "Absent", "Absent", "Absent", "Absent", "Present",
        "Absent", "Absent", "Present"
    ]
}

df = pd.DataFrame(data)
df["attendance_date"] = pd.to_datetime(df["attendance_date"])

# Function to find the latest absence streak
def find_absence_streaks(df):
    result = []

    for student_id, group in df.groupby("student_id"):
        group = group.sort_values("attendance_date").reset_index(drop=True)

        start_date = None
        prev_date = None
        absence_days = 0
        max_streak = {"start_date": None, "end_date": None, "days": 0}

        for _, row in group.iterrows():
            if row["status"] == "Absent":
                if start_date is None:
                    start_date = row["attendance_date"]
                    absence_days = 1
                elif prev_date and (row["attendance_date"] - prev_date).days == 1:
                    absence_days += 1
                else:
                    start_date = row["attendance_date"]
                    absence_days = 1

                if absence_days > max_streak["days"]:
                    max_streak = {"start_date": start_date, "end_date": row["attendance_date"], "d

            prev_date = row["attendance_date"]
```

```
            if max_streak["days"] > 3:
                result.append([student_id, max_streak["start_date"], max_streak["end_date"], max_strea

        return pd.DataFrame(result, columns=["student_id", "absence_start_date", "absence_end_date", "

    # Getting the result
    df_streaks = find_absence_streaks(df)
    print(df_streaks)
```

```
   student_id absence_start_date absence_end_date  total_absent_days
0         101         2024-01-01       2024-01-05                  5
1         103         2024-03-01       2024-03-08                  8
```

final step:

```
import pandas as pd
import re
from tabulate import tabulate

# Sample attendance streak data from Step 1
df_streaks = pd.DataFrame({
    "student_id": [101, 102, 103],
    "absence_start_date": ["2024-01-01", "2024-02-01", "2024-03-05"],
    "absence_end_date": ["2024-01-04", "2024-02-04", "2024-03-09"],
    "total_absent_days": [4, 4, 5]
})

df_streaks["absence_start_date"] = pd.to_datetime(df_streaks["absence_start_date"])
df_streaks["absence_end_date"] = pd.to_datetime(df_streaks["absence_end_date"])

# Sample student data
students = pd.DataFrame({
    "student_id": [101, 102, 103, 104],
    "student_name": ["Alice Johnson", "Bob Smith", "Charlie Brown", "David Lee"],
    "parent_email": ["alice.parent@example.com", "bob.parent@example.com", "invalid_email.com", "in
})

# Function to validate emails
def is_valid_email(email):
    pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
    return bool(re.match(pattern, email))

# Merge attendance streaks with student data
df_merged = df_streaks.merge(students, on="student_id", how="left")

# Validate emails
df_merged["valid_email"] = df_merged["parent_email"].apply(is_valid_email)

# Generate messages for valid emails
df_merged["msg"] = df_merged.apply(lambda row: f"Dear Parent, Your child {row['student_name']} was

# Keep only required columns
df_final = df_merged[["student_id", "absence_start_date", "absence_end_date", "total_absent_days",

# Filter out students with invalid emails
df_final = df_final[df_final["msg"] != ""]

# Print output in tabular format
print(tabulate(df_final, headers='keys', tablefmt='grid'))
```

	student_id	absence_start_date	absence_end_date	total_absent_days	parent_email	msg
0	101	2024-01-01 00:00:00	2024-01-04 00:00:00	4	alice.parent@example.com	Dear Parent, Your ch
1	102	2024-02-01 00:00:00	2024-02-04 00:00:00	4	bob.parent@example.com	Dear Parent, Your ch