In [ ]:
```
Nurshanov Dias IT3-2208
python 9 lab & practice
```

In [9]:
```
"""
9 lab
"""
import numpy as np
import pandas as pd

users = {
    'user_id': [1, 1, 2, 2, 3, 3, 4, 1],
    'action': ['Start', 'Cancel', 'Start', 'Publish', 'Start', 'Cancel', 'Start', 'Publish'],
    'action_date': ['2020-02-12', '2020-02-13', '2020-02-11', '2020-02-14', '2020-02-15', '2020-02-15', '2020-02

u_df = pd.DataFrame(users)
u_df['action_date'] = pd.to_datetime(u_df['action_date'])

result = (
    u_df.groupby('user_id', group_keys=False)
    .apply(lambda x: x.iloc[-1]['action_date'] - x.iloc[-2]['action_date'] if len(x) > 1 else pd.Timedelta(0), 
    .reset_index(name='time_elapsed'))

result = result[result['user_id'].isin([1, 2])]
result = result.sort_values(by='user_id')
print(result)


### 2 ###

projects = {
    'task_id': [1, 2, 3, 4, 5, 6, 7],
    'start_date': ['2020-10-01', '2020-10-02', '2020-10-03', '2020-10-13', '2020-10-14', '2020-10-28', '2020-10
    'end_date': ['2020-10-02', '2020-10-03', '2020-10-04', '2020-10-14', '2020-10-15', '2020-10-29', '2020-10-3
}

df = pd.DataFrame(projects)
df['start_date'] = pd.to_datetime(df['start_date'])
df['end_date'] = pd.to_datetime(df['end_date'])
df['project_id'] = (df['start_date'] > df['end_date'].shift(1) + pd.Timedelta(days=1)).cumsum()
result = (
    df.groupby('project_id')
    .agg(project_start=('start_date', 'min'),
         project_end=('end_date', 'max'),
         project_duration=('start_date', lambda x: (x.max() - x.min()).days))  # Убираем +1
    .reset_index(drop=True)
)
result = result.sort_values(by=['project_duration', 'project_start'], ascending=[True, False])
print(result)


### 3 ###

attendance_data = {
    'student_id': [1, 2, 3, 1, 2, 3, 1, 2, 3, 4],
    'school_date': ['4-3-20', '4-3-20', '4-3-20', '4-4-20', '4-4-20', '4-4-20', '4-5-20', '4-5-20', '4-5-20', '
    'attendance': [0, 1, 1, 1, 1, 1, 0, 1, 1, 1]
}
students_data = {
    'student_id': [1, 2, 3, 4],
    'school_id': [2, 1, 1, 2],
    'grade_level': [5, 4, 3, 4],
    'date_of_birth': ['4-3-12', '4-4-13', '4-5-14', '4-3-13']
}

a_df = pd.DataFrame(attendance_data)
s_df = pd.DataFrame(students_data)

a_df['school_date'] = pd.to_datetime(a_df['school_date'], format='%m-%d-%y')
s_df['date_of_birth'] = pd.to_datetime(s_df['date_of_birth'], format='%m-%d-%y')

s_df['birth_month_day'] = s_df['date_of_birth'].dt.strftime('%m-%d')
a_df['school_month_day'] = a_df['school_date'].dt.strftime('%m-%d')

merged_df = pd.merge(a_df, s_df, on='student_id')

merged_df['attended_on_birthday'] = (merged_df['attendance'] == 1) & (merged_df['school_month_day'] == merged_d
total_students = s_df['student_id'].nunique()

students_with_birthday_attendance = merged_df.groupby('student_id')['attended_on_birthday'].max().sum()
fraction = round(students_with_birthday_attendance / total_students, 2)
print("Fraction of students who attended school on their birthday:", fraction)
```

```
### 4 ###

users_data = {
    'user_id': [1, 2, 3, 4, 5, 6, 7, 8],
    'join_date': ['01-01-20', '01-10-20', '02-05-20', '02-12-20', '02-25-20', '03-01-20', '03-01-20', '03-04-20
    'invited_by': [0, 1, 2, 3, 2, 0, 4, 7]
}

users_df = pd.DataFrame(users_data)
users_df['join_date'] = pd.to_datetime(users_df['join_date'], format='%m-%d-%y')
merged_df = pd.merge(
    users_df,
    users_df,
    left_on='user_id',
    right_on='invited_by',
    suffixes=('_inviter', '_invitee')
)
merged_df['cycle_time'] = (merged_df['join_date_invitee'] - merged_df['join_date_inviter']).dt.days
merged_df['inviter_month'] = merged_df['join_date_inviter'].dt.month
average_cycle_time = (
    merged_df.groupby('inviter_month')['cycle_time']
    .mean()
    .reset_index(name='avg_cycle_time')
    .sort_values(by='inviter_month')
)
print(average_cycle_time)
```

```
   user_id time_elapsed
0        1       6 days
1        2       3 days
  project_start project_end  project_duration
1    2020-10-13  2020-10-15                 1
2    2020-10-28  2020-10-31                 2
0    2020-10-01  2020-10-04                 2
Fraction of students who attended school on their birthday: 0.5
   inviter_month  avg_cycle_time
0              1            27.0
1              2            12.5
2              3             3.0
```

In [10]:

```
"""
9 practice
"""
import pandas as pd

date_obj = pd.Timestamp('2012-12-12')
print(date_obj)

datetime_obj = pd.Timestamp('2012-12-12 15:24:00')
print(datetime_obj)

local_datetime = pd.Timestamp.now()
print(local_datetime)

date_only = pd.Timestamp('2012-12-12').date()
print(date_only)

time_only = pd.Timestamp('15:24:00').time()
print(time_only)

current_date = pd.Timestamp.today().date()
print(current_date)

current_time = pd.Timestamp.now().time()
print(current_time)

current_date = pd.Timestamp.today()
first_day_three_months_ago = (current_date - pd.DateOffset(months=3)).replace(day=1)
print(first_day_three_months_ago)

nauryz = pd.Timestamp('2021-03-22')
day_before = nauryz - pd.Timedelta(days=1)
day_after = nauryz + pd.Timedelta(days=1)
print("Day before Nauryz:", day_before.date())
print("Day after Nauryz:", day_after.date())

start_date = pd.Timestamp('2021-03-22')
end_date = pd.Timestamp('2021-03-26')
working_days = pd.date_range(start=start_date, end=end_date, freq='B')
print(f"Working days between {start_date.date()} and {end_date.date()}: {len(working_days)}")
```

```
2012-12-12 00:00:00
2012-12-12 15:24:00
2024-12-02 23:05:37.151154
2012-12-12
15:24:00
2024-12-02
23:05:37.152192
2024-09-01 23:05:37.152284
Day before Nauryz: 2021-03-21
Day after Nauryz: 2021-03-23
Working days between 2021-03-22 and 2021-03-26: 5
```

In [ ]: