## 10. Project Aggregation

The projects table contains three columns: task\_id, start\_date, and end\_date. The difference between end\_date and start\_date is 1 day for each row in the table. If task end dates are consecutive they are part of the same project. Projects do not overlap. Write a query to return the start and end dates of each project, and the number of days it took to complete. Order by ascending project duration, and ascending start date in the case of a tie.

```
import pandas as pd
import numpy as np
data = {'task_id'
                     : [1,2,3,4,5,6,7],
        'start_date' : ['01-0CT-20',
                         '02-0CT-20',
                         '03-0CT-20',
                         '13-0CT-20',
                         '14-0CT-20'
                         '28-OCT-20',
                         '30-0CT-20'],
                     : ['02-0CT-20',
         'end date'
                         '03-0CT-20',
                         '04-0CT-20',
                         '14-0CT-20',
                         '15-0CT-20',
                         '29-0CT-20',
                         '31-0CT-20']
        }
projects = pd.DataFrame(data)
projects['start_date']=(pd.to_datetime(projects['start_date']
                                        ,format="%d-%b-%y"
)
projects['end_date']=(pd.to_datetime(projects['end_date']
                                      ,format="%d-%b-%y"
                          )
print(projects)
\overline{2}
        task_id start_date
                               end_date
               1 2020-10-01 2020-10-02
    0
    1
               2 2020-10-02 2020-10-03
    2
              3 2020-10-03 2020-10-04
    3
              4 2020-10-13 2020-10-14
    4
              5 2020-10-14 2020-10-15
    5
              6 2020-10-28 2020-10-29
              7 2020-10-30 2020-10-31
```

```
df1=(projects.sort_values('start_date')
             .assign(project_id=lambda x:
                                        x['start_date']!=
                                        x['end date'].shift(1)
                                       .cumsum()
              )
)
print(f'Identifying unique projects:\n{df1}')
    Identifying unique projects:
       task_id start_date end_date project_id
              1 2020-10-01 2020-10-02
    0
              2 2020-10-02 2020-10-03
    1
                                                   1
              3 2020-10-03 2020-10-04
    2
                                                   1
    3
              4 2020-10-13 2020-10-14
                                                   2
              5 2020-10-14 2020-10-15
                                                   2
    4
    5
                                                   3
              6 2020-10-28 2020-10-29
              7 2020-10-30 2020-10-31
    6
                                                   4
df2=(df1.groupby('project_id')
        .agg(project_start=('start_date'
                            ,'min'
             ,project_end=('end_date'
                           ,'max'
              )
         )
)
print(f'Returning starts and completions:\n{df2}')
    Returning starts and completions:
                project_start project_end
    project_id
    1
                   2020-10-01
                                2020-10-04
    2
                   2020-10-13 2020-10-15
    3
                   2020-10-28 2020-10-29
    4
                   2020-10-30 2020-10-31
timeline_projects=(projects.sort_values('start_date')
                           .assign(project_id=lambda x:
                                                     x['start_date']!=
                                                     x['end_date'].shift(1)
                                                     .cumsum()
                            )
                           .groupby('project_id'
                                    ,as_index=False
                           .agg(project_start=('start_date',
                                ),
                                project_end=('end_date',
                                             'max'
```

```
)
                              )
                             .assign(project_duration=lambda x:
                                                       (x['project_end']-x['project_start']).d
)
timeline_projects.sort_values(['project_duration','project_start'])
\overline{2}
         project_id project_start project_end project_duration
     2
                   3
                           2020-10-28
                                          2020-10-29
                                                                           ılı
      3
                   4
                           2020-10-30
                                          2020-10-31
                                                                       1
      1
                   2
                           2020-10-13
                                          2020-10-15
                                                                       2
```

2020-10-04

3

0

1

2020-10-01