06. Content recommendation.

Using the following two tables, write a query to return page recommendations to a social media user based on the pages that their friends have liked, but that they have not yet marked as liked. Order the result by ascending user ID.

```
import pandas as pd
import numpy  as np

data1 = {'user_id': [1,1,1,2,3,3,4,4],
         'friend'  : [2,3,4,1,1,4,1,3]
        }

data2 = {'user_id' : [1,1,1,2,3,3,4],
         'page_likes' : ['A','B','C','A','B','C','B']
        }

friends = pd.DataFrame(data1)
likes   = pd.DataFrame(data2)


print(friends)
```

```
   user_id  friend
0        1       2
1        1       3
2        1       4
3        2       1
4        3       1
5        3       4
6        4       1
7        4       3
```

```
print(likes)
```

```
   user_id page_likes
0        1          A
1        1          B
2        1          C
3        2          A
4        3          B
5        3          C
6        4          B
```

```
df1=(pd.merge(friends
              ,likes.rename(columns={
                                     'user_id':'friend'
                                     ,'page_likes':'recommendation'
                                    }
                            )
              ,on='friend'
```

```
              ,how='inner'
        )
       .drop(columns=['friend'])
)
print(df1)
```

```
    user_id recommendation
0         1              A
1         1              B
2         1              C
3         1              B
4         2              A
5         2              B
6         2              C
7         3              A
8         3              B
9         3              C
10        3              B
11        4              A
12        4              B
13        4              C
14        4              B
15        4              C
```

```
df2=(pd.merge(friends
              ,likes.rename(columns={'user_id':'friend'})
              ,on='friend'
        )
)
df3=(pd.merge(df2
              ,likes
              ,on=['user_id','page_likes']
              ,how='left'
              ,indicator=True
        )
       .rename(columns={'page_likes':'recommendation'})
)
print(df3)
```

```
    user_id  friend recommendation     _merge
0         1       2              A       both
1         1       3              B       both
2         1       3              C       both
3         1       4              B       both
4         2       1              A       both
5         2       1              B  left_only
6         2       1              C  left_only
7         3       1              A  left_only
8         3       1              B       both
9         3       1              C       both
10        3       4              B       both
11        4       1              A  left_only
12        4       1              B       both
13        4       1              C  left_only
14        4       3              B       both
15        4       3              C  left_only
```

```
df=(pd.merge(friends
             ,likes.rename(columns={'user_id':'friend'})
             ,on='friend'
        )
)
recommendations=(pd.merge(df
                          ,likes
                          ,on=['user_id','page_likes']
                          ,how='left'
                          ,indicator=True
                   )
                 .rename(columns={'page_likes':'recommendation'})
                 .query("_merge=='left_only'")[['user_id','recommendation']]
                 .drop_duplicates()
                 .sort_values(by='user_id')
)
recommendations
```

	user_id	recommendation
5	2	B
6	2	C
7	3	A
11	4	A
13	4	C

Next steps: Generate code with recommendations | View recommended plots | New interactive sheet