

✓ 01. Cancellation Rates

From the following table of user IDs, actions, and dates, write a query to return the publication and cancellation rate for each user.

```
# pip install pandas
# pip install numpy
# pip install SQLAlchemy
# pip install cx_Oracle

import pandas as pd
import numpy as np
# import cx_Oracle
# import sqlalchemy
# from sqlalchemy.exc import SQLAlchemyError

'''
01. Cancellation Rates.
```

Writing a query to return the publication and cancellation rate for each user

```
try:
    engine = sqlalchemy.create_engine("oracle+cx_Oracle://usr:pswd@localhost/?service_name=orclpdb1",
    table = """select * from users_p1""";
    users = pd.read_sql(table, engine)
    users

except SQLAlchemyError as e:
    print(e)
'''
```

➡ '\n01. Cancellation Rates.\n\nWriting a query to return the publication and cancellation\nrate for each user\n\n\ntry:\n engine = sqlalchemy.create_engine("oracle+cx_Oracle://usr:pswd@localhost/?service_name=orclpdb1", arraysize=1000)\n\n table = """select * from users_p1""";\n users = pd.read_sql(table, engine)\n users\n\nexcept SQLAlchemyError as e:\n print(e)\n'

```
data = {'user_id' : [1,1,2,1,1,2,3,3,4],
        'action'  : ['start','cancel','start',
                     'start','publish','publish',
                     'start','cancel','start'],
        'dates'   : ['01-JAN-20',
                     '02-JAN-20',
```

```

        '03-JAN-20',
        '03-JAN-20',
        '04-JAN-20',
        '04-JAN-20',
        '05-JAN-20',
        '06-JAN-20',
        '07-JAN-20']
    }

```

```

users = pd.DataFrame(data)
print(users.head(5))

```

```

⇒ user_id  action  dates
0         1   start 01-JAN-20
1         1  cancel 02-JAN-20
2         2   start 03-JAN-20
3         1   start 03-JAN-20
4         1 publish 04-JAN-20

```

```

df1 = pd.get_dummies(users['action'])
print(df1.head(5))

```

```

⇒ cancel  publish  start
0   False    False   True
1    True    False  False
2   False    False   True
3   False    False   True
4   False     True  False

```

```

df2 = (pd.get_dummies(users['action'])
       .groupby(users['user_id'])
       .sum()
       )
print(df2.head(5))

```

```

⇒ user_id  cancel  publish  start
1         1         1         2
2         0         1         1
3         1         0         1
4         0         0         1

```

```

actions = (pd.get_dummies(users['action'])
           .groupby(users['user_id'])
           .sum()
           .assign(publish_rate = lambda x : x['publish'] / x['start'],
                   cancel_rate = lambda x : x['cancel'] / x['start']
           )
           .replace(np.inf, 0)
           .reset_index()
           )

actions[['user_id', 'publish_rate', 'cancel_rate']]

```



	user_id	publish_rate	cancel_rate
0	1	0.5	0.5
1	2	1.0	0.0
2	3	0.0	1.0
3	4	0.0	0.0