

HOMEWORK 10

Using our EcommerceDB , build a recommendation rest service that takes in a customer as input and returns the item for which a promotion can be offered for that customer.

Your recommendation model can be as simple as finding the most freq bought item by that customer or anything else you choose.

Turn in your HW with the codebase and a small write describing your model.

In order to figure out which items should be given promotion discount for one specific customer, the most frequently bought items by this customer are considered with priority. This question can be addressed with MySQL database (EcommerceDB) driven rest service API in Python with Flask.

First, the Customer ID can be figured out by Customer Name and other supporting information (such as Zip in Address table, or telephone number which is not listed here), especially for some customer names may have more than one Customer IDs, or same Customer Name and Customer ID with multiple Zips. For simplification, here just consider the shop manager can directly identify the coming Customer ID with Customer Name. For example, “Ed Laws” has Customer ID 51 after checking other reference information.

Second, use the Customer ID to search for the most frequently bought item by this customer and provide promotion discount. For “Ed Laws” with Customer ID 51, we can consider giving him promotion discount for items with Item Numbers 87, 59, 12, 35,

Below are the details for the codes and outputs for this rest service APIs:

1. Codes for recommendation rest service (with screenshot)

```
#!/flask/bin/python
from flask import Flask, render_template
import pymysql
from flask import jsonify
from flask import Flask, abort, request
import logging
```

```

app = Flask(__name__)

@app.route('/')
def index():
    return "Promotion item list for specific customer"

#### Rest Service definition for returning the CustomerID from CustomerTable
@app.route('/api/v1.0/getCustomerID', methods=['GET'])
def getCustomerID():
    global query_result;
    try:
        # Create connection to the MYSQL instance running on the local machine

        logging.warning("Inside getCustomerID()")
        connection = pymysql.Connection(host='localhost', user='root',
passwd='Newlife@2018', db='EcommerceDB')
        cursor = connection.cursor()
        logging.warning("Connection with DB established")
        cursor.execute("SELECT CustomerID, CustomerName from CustomerTable order by
CustomerName, CustomerID;")
        query_result = [ dict(line) for line in [zip([ column[0] for column in
cursor.description], row) for row in cursor.fetchall() ] ]
    except Exception as e:
        print ("Error [%r]" % (e))
        #sys.exit(1)
    finally:
        if cursor:
            cursor.close()

    ## Return the list of actors as a json object
    return jsonify({'CustomerName': query_result})

#### Rest Service definition for specific CustomerID.
@app.route('/api/v1.0/getPromotionDetails/<CustomerID>', methods=['POST', 'GET'])
def PromotionDetails(CustomerID):
    global query_result
    try:
        # Create connection to the MYSQL instance running on the local machine
        # Change the credentials to match your system
        connection = pymysql.connect(
            host='localhost', user='root', passwd='Newlife@2018', db='EcommerceDB')
        cursor = connection.cursor()
        logging.warning("Connection established")
        # Read the input json from the incoming request
        reqObj = request.json
        logging.warning("Request Object \n %s" % reqObj)

        queryStr = "with B as ( select ct.CustomerID , ct.CustomerName, ol.Quantity,
ol.Item_Number as ItemNumber from CustomerTable as ct, OrderHeader as oh, OrderLine as
ol \
                        where ct.CustomerID = oh.CustomerTable_CustomerID and oh.OrderNumber =
ol.OrderHeader_OrderNumber ) \
                        select B.* from B, (select max(Quantity) as MaxFreq, ItemNumber from B
group by ItemNumber desc) as A \
                        where A.ItemNumber=B.ItemNumber and A.MaxFreq =B.Quantity and

```

```

CustomerId= %d;" %int(CustomerID)

    logging.warning(queryStr)
    cursor.execute(queryStr)
    query_result = [dict(line) for line in [zip(
        [column[0] for column in cursor.description], row) for row in
cursor.fetchall()]]
except Exception as e:
    logging.warning("Error [%r]" % (e))
    # sys.exit(1)
finally:
    if cursor:
        cursor.close()

    return jsonify({'PromotionDetails': query_result})

## Starts the server for serving Rest Services
if __name__ == '__main__':
    app.run(debug=False)

```

Screenshot of the codes

```

1  #!/usr/bin/python
2  from flask import Flask, render_template
3  import pymysql
4  from flask import jsonify
5  from flask import Flask, abort, request
6  import logging
7
8
9
10 app = Flask(__name__)
11
12
13 @app.route('/')
14 def index():
15     return "Promotion item list for specific customer"
16
17
18 ### Rest Service definition for returning the CustomerID from CustomerTable
19 @app.route('/api/v1.0/getCustomerID', methods=['GET'])
20 def getCustomerID():
21     global query_result
22     try:
23         # Create connection to the MYSQL instance running on the local machine
24         logging.warning("Inside getCustomerID()")
25         connection = pymysql.connect(host='localhost', user='root', passwd='NewLife@2018', db='EcommerceDB')
26         cursor = connection.cursor()
27         logging.warning("Connection with DB established")
28         cursor.execute("SELECT CustomerID, CustomerName from CustomerTable order by CustomerName, CustomerID;")
29         query_result = [ dict(line) for line in [zip([ column[0] for column in cursor.description], row) for row in cursor.fetchall() ] ]
30     except Exception as e:
31         print ("Error [%r]" % (e))
32         sys.exit(1)
33     finally:
34         if cursor:
35             cursor.close()
36
37     ## Return the list of actors as a json object
38     return jsonify({'CustomerName': query_result})
39
40
41
42 ### Rest Service definition for specific CustomerID.
43 @app.route('/api/v1.0/getPromotionDetails/<CustomerID>', methods=['POST', 'GET'])
44 def PromotionDetails(CustomerID):
45     global query_result
46     try:
47         # Create connection to the MYSQL instance running on the local machine
48         # Change the credentials to match your system
49         connection = pymysql.connect(
50             host='localhost', user='root', passwd='NewLife@2018', db='EcommerceDB')
51         cursor = connection.cursor()
52         logging.warning("Connection established")
53         # Read the input json from the incoming request
54         reqObj = request.json
55         logging.warning("Request Object \n %s" % reqObj)
56
57         queryStr = "with B as ( select ct.CustomerID, ct.CustomerName, ol.Quantity, ol.Item_ItemNumber as ItemNumber from CustomerTable as ct, OrderHeader as oh, OrderLine as ol \
58             where ct.CustomerID = oh.CustomerTable_CustomerID and oh.OrderNumber = ol.OrderHeader_OrderNumber ) \
59             select B.* from B, (select max(Quantity) as MaxFreq, ItemNumber from B group by ItemNumber desc) as A \
60             where A.ItemNumber=B.ItemNumber and A.MaxFreq =B.Quantity and CustomerID= %d;" %int(CustomerID)
61
62         logging.warning(queryStr)
63         cursor.execute(queryStr)
64         query_result = [dict(line) for line in [zip(
65             [column[0] for column in cursor.description], row) for row in cursor.fetchall()]]
66     except Exception as e:
67         logging.warning("Error [%r]" % (e))
68         # sys.exit(1)
69     finally:
70         if cursor:
71             cursor.close()
72
73     return jsonify({'PromotionDetails': query_result})
74
75 ## Starts the server for serving Rest Services
76 if __name__ == '__main__':
77     app.run(debug=False)

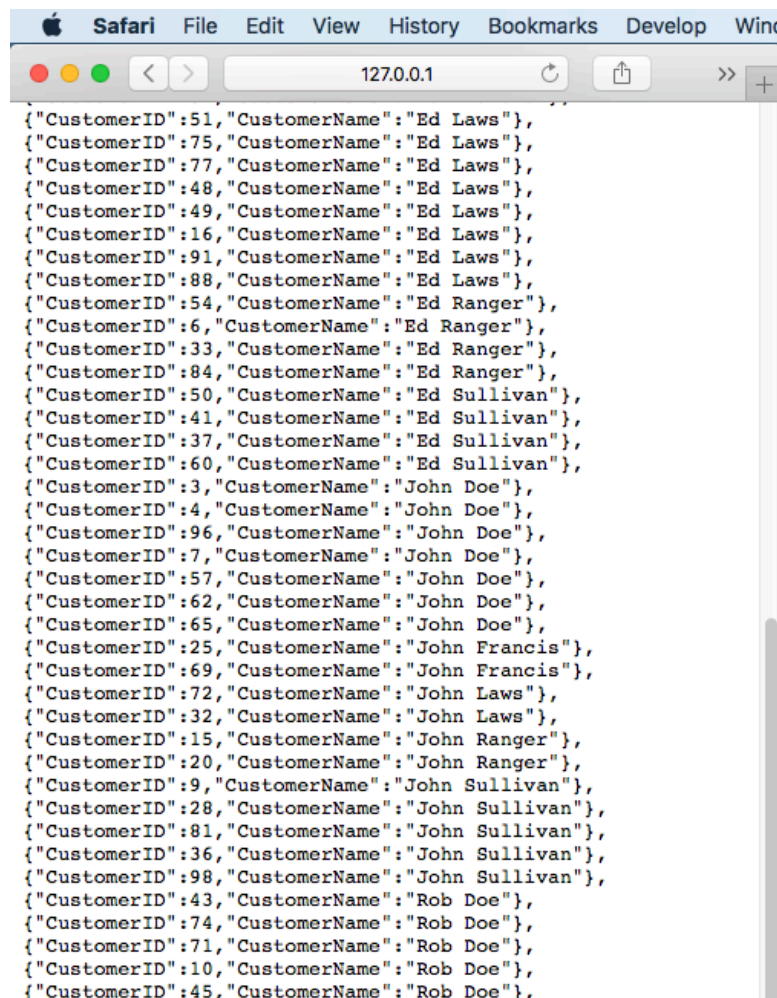
```

Terminal confirmation

```
Last login: Sat Jul 21 17:18:44 on ttys000
-bash: Export: command not found
Shon@rpgs-MBP:~/Desktop/shon@rpgs$ python mainlib.py
* Serving Flask app "mainlib" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Jul/2018 17:21:35] "GET / HTTP/1.1" 200 -
WARNING:root:Connection with DB established
127.0.0.1 - - [21/Jul/2018 17:22:04] "GET /api/v1.0/getCustomerID HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [21/Jul/2018 17:22:04] "GET /api/v1.0/getCustomerID HTTP/1.1" 200 -
WARNING:root:Inside getCustomerID()
WARNING:root:Connection with DB established
127.0.0.1 - - [21/Jul/2018 17:22:05] "GET /api/v1.0/getCustomerID HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [21/Jul/2018 17:22:05] "GET /api/v1.0/getCustomerID HTTP/1.1" 200 -
WARNING:root:Connection established
WARNING:root:Request Object
None
WARNING:root:with 8 as ( select ct.CustomerID , ct.CustomerName, ol.Quantity, ol.ItemNumber as ItemNumber from CustomerTable as ct, OrderHeader as oh, OrderLine as ol
                                where ct.CustomerID = oh.CustomerTable_CustomerID and oh.OrderNumber = ol.Order
                                Header_OrderNumber )
                                select B.* from 8, (select max(Quantity) as MaxFreq, ItemNumber from 8 group by ItemNumber desc) as A
                                where A.ItemNumber=8.ItemNumber and A.MaxFreq =8.Quantity and CustomerId= 51;
127.0.0.1 - - [21/Jul/2018 17:22:28] "GET /api/v1.0/getPromotionDetails/51 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [21/Jul/2018 17:22:28] "GET /api/v1.0/getPromotionDetails/51 HTTP/1.1" 200 -
WARNING:root:Connection established
WARNING:root:Request Object
None
WARNING:root:with 8 as ( select ct.CustomerID , ct.CustomerName, ol.Quantity, ol.ItemNumber as ItemNumber from CustomerTable as ct, OrderHeader as oh, OrderLine as ol
                                where ct.CustomerID = oh.CustomerTable_CustomerID and oh.OrderNumber = ol.Order
                                Header_OrderNumber )
                                select B.* from 8, (select max(Quantity) as MaxFreq, ItemNumber from 8 group by ItemNumber desc) as A
                                where A.ItemNumber=8.ItemNumber and A.MaxFreq =8.Quantity and CustomerId= 51;
127.0.0.1 - - [21/Jul/2018 17:22:29] "GET /api/v1.0/getPromotionDetails/51 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [21/Jul/2018 17:22:29] "GET /api/v1.0/getPromotionDetails/51 HTTP/1.1" 200 -
```

2. Get CustomerID with CustomerName

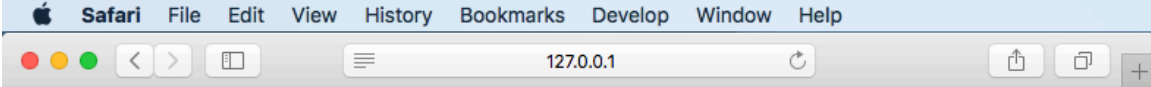
For example, “Ed Laws” has CustomerID 51 after checking other information



```
{\"CustomerID\": 51, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 75, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 77, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 48, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 49, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 16, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 91, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 88, \"CustomerName\": \"Ed Laws\"},
{\"CustomerID\": 54, \"CustomerName\": \"Ed Ranger\"},
{\"CustomerID\": 6, \"CustomerName\": \"Ed Ranger\"},
{\"CustomerID\": 33, \"CustomerName\": \"Ed Ranger\"},
{\"CustomerID\": 84, \"CustomerName\": \"Ed Ranger\"},
{\"CustomerID\": 50, \"CustomerName\": \"Ed Sullivan\"},
{\"CustomerID\": 41, \"CustomerName\": \"Ed Sullivan\"},
{\"CustomerID\": 37, \"CustomerName\": \"Ed Sullivan\"},
{\"CustomerID\": 60, \"CustomerName\": \"Ed Sullivan\"},
{\"CustomerID\": 3, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 4, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 96, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 7, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 57, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 62, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 65, \"CustomerName\": \"John Doe\"},
{\"CustomerID\": 25, \"CustomerName\": \"John Francis\"},
{\"CustomerID\": 69, \"CustomerName\": \"John Francis\"},
{\"CustomerID\": 72, \"CustomerName\": \"John Laws\"},
{\"CustomerID\": 32, \"CustomerName\": \"John Laws\"},
{\"CustomerID\": 15, \"CustomerName\": \"John Ranger\"},
{\"CustomerID\": 20, \"CustomerName\": \"John Ranger\"},
{\"CustomerID\": 9, \"CustomerName\": \"John Sullivan\"},
{\"CustomerID\": 28, \"CustomerName\": \"John Sullivan\"},
{\"CustomerID\": 81, \"CustomerName\": \"John Sullivan\"},
{\"CustomerID\": 36, \"CustomerName\": \"John Sullivan\"},
{\"CustomerID\": 98, \"CustomerName\": \"John Sullivan\"},
{\"CustomerID\": 43, \"CustomerName\": \"Rob Doe\"},
{\"CustomerID\": 74, \"CustomerName\": \"Rob Doe\"},
{\"CustomerID\": 71, \"CustomerName\": \"Rob Doe\"},
{\"CustomerID\": 10, \"CustomerName\": \"Rob Doe\"},
{\"CustomerID\": 45, \"CustomerName\": \"Rob Doe\"},
```

3. Use CustomerID to get the most frequent ItemNumber for promotion recommendation

For “Ed Laws” with “CustomerID” = 51, we can consider giving promotions for these items which were most frequently bought previously, such as: ItemNumber 87, 59, 12, 35, 71, ... (as listed below)



```
{
  "PromotionDetails": [
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 26,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 87,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 59,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 12,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 35,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 71,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 68,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 88,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 81,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 70,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 12,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 61,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 50,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 82,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 84,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 13,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 41,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 32,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 4,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 3,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 52,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 93,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 34,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 48,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 72,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 34,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 98,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 6,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 42,
      "Quantity": 10
    },
    {
      "CustomerID": 51,
      "CustomerName": "Ed Laws",
      "ItemNumber": 90,
      "Quantity": 10
    }
  ]
}
```