Pharmacy Drugs Inventory Management

Milestone: Python Application

Group 11

Sai Varun Kumar Namburi

FNU Meenal

857-265-1349 (Sai Varun)

848-667-4233 (Meenal)

namburi.sai@northeastern.edu

lnu.meenal@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Sai Varun

Signature of Student 2: Meenal

Submission Date: November 5, 2022

Python Application:

This milestone aims to connect the MySQL database with python and perform basic analysis.

I have used the "MySQLdb" library in python, which is used for connecting to the MySQL database
Here you can have a look at how I connected to my local database.

```
In 2
#!/usr/bin/python
import MySQLdb
import pandas as pd
# Connect
db = MySQLdb.connect(host="localhost",
                     user="root",
                     passwd="admin",
                     db="inventory")
```

Connecting to a database requires 4 attributes, username, password, dbname, and hostname.
After connecting to a database now we can start using python functionalities to query the database and retrieve the data.

Basic Analysis:

1. Checking all the orders which have a total amount of more than 25000 and are ordered by total_amount in descending order
   Query: **select * from orders where total_amount>25000 order by total_amount desc;**

Output for the above query after running in the python, it returns as a data frame

```
In 18
# Execute SQL select statement
#1. Checking the all the orders which have total amount more than 25000 and ordered by total_amount in descending order
sql = "select * from inventory.orders where total_amount>25000 order by total_amount desc"
df_1 = pd.read_sql(sql, db)
print(df_1)
```

	order_id	pharmacy_id	shipment_id	order_date	payment_type	total_amount
0	896	170	3937	2021-05-17	Online	49763
1	712	628	3798	2021-03-01	Online	49702
2	662	809	2738	2022-07-22	Online	48800
3	135	976	9827	2021-12-22	Online	48684
4	519	263	1588	2021-04-20	Online	47932
5	723	827	5536	2022-01-12	Offline	47915
6	683	304	9325	2022-07-08	Online	47446
7	956	891	5420	2021-05-29	Online	47285
8	457	344	4544	2022-01-06	Offline	46736
9	694	610	9535	2022-06-16	Online	46364

2. Now I have added one more filter condition to check orders for the year 2021
   Query: **select * from orders where total_amount>25000 and order_date< '2021-12-31' order by total_amount desc;**

   Output:

```
In 22
#2. Now I have added one more filter condition to check orders of year 2021
sql = "select * from orders where total_amount>25000 and order_date<'2021-12-31' order by total_amount desc"
df_2 = pd.read_sql(sql, db)
print(df_2)
```

	order_id	pharmacy_id	shipment_id	order_date	payment_type	total_amount
0	896	170	3937	2021-05-17	Online	49763
1	712	628	3798	2021-03-01	Online	49702
2	135	976	9827	2021-12-22	Online	48684
3	519	263	1588	2021-04-20	Online	47932
4	956	891	5420	2021-05-29	Online	47285
5	998	720	7772	2021-08-31	Offline	45954
6	358	132	2046	2021-04-27	Online	43158
7	352	241	8297	2021-11-08	Online	42617
8	967	590	9422	2021-05-27	Offline	42051
9	359	154	3154	2021-10-05	Online	40702

3. Now we can find the data which have a total amount of t less than 25000 and have an order date before Jan 2022
   Query: **select * from orders join order_details on orders.order_id = order_details.order_id where orders.total_amount>25000 and orders.order_date < ' 2021-12-31' order by orders.total_amount desc;**

Output:

```
In 23
#3. The below query returns the orders which are ordered after Dec 2021 and having total amount is greater than 25000
sql = "select * from orders join order_details on orders.order_id=order_details.order_id where orders.total_amount>25000
  and orders.order_date<'2021-12-31' order by orders.total_amount desc"
df_3 = pd.read_sql(sql, db)
print(df_3)
```

```
     order_id  pharmacy_id  shipment_id order_date payment_type  total_amount  \
0         896          170         3937 2021-05-17       Online         49763
1         712          628         3798 2021-03-01       Online         49702
2         135          976         9827 2021-12-22       Online         48684
3         519          263         1588 2021-04-20       Online         47932
4         956          891         5420 2021-05-29       Online         47285
5         998          720         7772 2021-08-31      Offline         45954
6         358          132         2046 2021-04-27       Online         43158
7         352          241         8297 2021-11-08       Online         42617
8         967          590         9422 2021-05-27      Offline         42051
9         359          154         3154 2021-10-05       Online         40702
```

4. Next, we are going to find the drugs which are having less than 200 stock in the store
   Query: **select * from stock_details where stock_left < 200;**
   Output:

```
In 24
#4. Which means we have less stock left for those drugs
sql = "select * from stock_details where stock_left <200"
df_4 = pd.read_sql(sql, db)
print(df_4)
```

```
     stock_id  drug_id  warehouse_id  stock_left last_ordered_date  \
0        1220       82            32         169        2022-02-25
1        2264       39            95         195        2022-04-12
2        2475       46            92          76        2022-04-15
3        2830       28            77          68        2022-10-13
4        2967       74            51          55        2022-04-29
5        3273       23            42          36        2021-12-07
6        3396       58            85         178        2022-06-04
7        3527        4            63         108        2022-07-02
8        4123       40            75          11        2022-05-03
9        4342       60            22         109        2022-01-14
```

5. Finding the order details of those who have purchased the drugs with less quantity less than 30
   Query:
   select od.order_id,od.drug_id,d.drug_name, od.quantity, o.order_date, o.payment_type, o.total_amount
   from orders o join order_details od on o.order_id=od.order_id

join drugs d on d.drug_id=od.drug_id where od.quantity<30;

Output:

```
In 4
#5. Finding the order details who have ordered the quantity less than 30
sql = "select od.order_id,d.drug_name, od.quantity, o.order_date, o.payment_type, o.total_amount \
from orders o join order_details od on o.order_id=od.order_id \
join drugs d on d.drug_id=od.drug_id \
where od.quantity<30;"
df_5 = pd.read_sql(sql, db)
print(df_5)
```

	order_id	drug_name	quantity	order_date	payment_type	total_amount
0	553	Anastrozole	21	2021-03-07	Offline	12236
1	513	AndroGel	29	2021-03-23	Offline	26605
2	358	Annovera	25	2021-04-27	Online	43158
3	112	Atenolol	26	2022-04-06	Online	23630
4	770	Histrelin	23	2022-05-06	Offline	33000
5	479	Hizentra	27	2022-05-15	Offline	24219
6	325	Humira	24	2022-06-21	Online	35526
7	901	Humulin R	20	2022-08-09	Online	32234
8	258	Levocetirizine	22	2021-02-12	Offline	15041
9	609	Levofloxacin	28	2021-03-14	Online	21349

Now after doing the analysis with the data, you must close the DB connection which has created at the start of the application

Output:

```
In 5
# Close the connection
db.close()
```