

Group 6
Lancer Guo, 260728557
Xintong Li, 260710002
Bessie Luo, 260708568
Zhuoran Zhao, 260716696

COMP 421 Project Deliverable 3

1. Stored procedure

This function(procedure) is used perform the action needed after customer is checking out, when customer selects to check out all the items in the cart and trying to pay, this function will be invoked to do the following:

- Then it will create a payment record in Payments table, with customer id filled, and generate a unique payment id for this payment. Tracking number is not filled at first because it is responsibility of the carrier to generate this number.
- Then it will move all the items from CartItems to OrderItems table combining with the new payment id
- At the same time it will update the detail info and total cost
- In the end, it will update detail info and total cost to the payment record in Payments
- Checks that if there are no cart items, then the customer cannot submit payment

```
CREATE OR REPLACE FUNCTION submit_payment (IN input_cid INT) RETURNS TEXT
AS $$
DECLARE new_pid INT;
DECLARE all_items_quan INT;
DECLARE r_quan INT;
DECLARE r_pname VARCHAR(30);
DECLARE r_bname VARCHAR(30);
DECLARE p_price DECIMAL(7, 2);
DECLARE cur_stock INT;
DECLARE total_cost INT DEFAULT 0;
DECLARE payment_detail VARCHAR(1000) DEFAULT 'Ordered Items: ';
DECLARE cursor_cart CURSOR FOR
SELECT quantity, pname, bname FROM CartItems WHERE cid = input_cid;
BEGIN
SELECT COUNT(*) INTO all_items_quan FROM CartItems WHERE cid = input_cid;
IF (all_items_quan <= 0)
THEN
RETURN 'No item in cart for this customer';
ELSE
--create a new payment and get its id
INSERT INTO Payments (cid) VALUES (input_cid) RETURNING pid INTO new_pid;
--select all the item in cart
OPEN cursor_cart;
LOOP
FETCH cursor_cart INTO r_quan, r_pname, r_bname;
EXIT WHEN NOT FOUND;

SELECT stock, price INTO cur_stock, p_price FROM ForSale WHERE pname = r_pname AND bname = r_bname;
INSERT INTO OrderItems (quantity, pname, bname, pid, cid) VALUES (r_quan, r_pname, r_bname, new_pid, input_cid);
payment_detail := payment_detail || r_pname || ' ' || r_bname || ' * ' || r_quan || ', ';
total_cost = total_cost + p_price; --calculate total cost
--delete entry
DELETE FROM CartItems WHERE CURRENT OF cursor_cart;
END LOOP;
CLOSE cursor_cart;
payment_detail := payment_detail || 'Total cost: ' || total_cost;
--update the payment detail about what the customer bought and the cost
UPDATE Payments SET orderDetail = payment_detail WHERE pid = new_pid;

RETURN 'Success';
END IF;
END
$$ LANGUAGE plpgsql;
```

Query statement before the stored procedure is executed:

```
[cs421=> SELECT * FROM CartItems WHERE cid = 531;
```

item_id	quantity	pname	bname	cid
38	20	Soy Face Cleanser	Fresh	531
39	2	Ultra Long Wear Foundation	Lancome	531
40	1	Rouge Dior Ultra Lipstick	Dior	531

(3 rows)

Query statement after the stored procedure is executed:

```
[cs421=> SELECT submit_payment(531);
```

```
submit_payment
-----
Success
(1 row)
```

```
[cs421=> SELECT * FROM cartitems WHERE cid = 531;
```

```
item_id | quantity | pname | bname | cid
-----+-----+-----+-----+-----
(0 rows)
```

```
[cs421=> SELECT * FROM orderitems WHERE cid = 531;
```

orderitem_id	quantity	pname	bname	pid	cid
12	2	Ultra Long Wear Foundation	Lancome	71	531
13	1	Rouge Dior Ultra Lipstick	Dior	71	531
14	3	Soy Face Cleanser	Fresh	72	531
15	2	Ultra Long Wear Foundation	Lancome	72	531
16	1	Rouge Dior Ultra Lipstick	Dior	72	531
17	3	Soy Face Cleanser	Fresh	73	531
18	2	Ultra Long Wear Foundation	Lancome	73	531
19	1	Rouge Dior Ultra Lipstick	Dior	73	531
11	1	Soy Face Cleanser	Fresh	71	531
21	1	Night Cream	Fresh	79	531
22	3	Soy Face Cleanser	Fresh	81	531
23	2	Ultra Long Wear Foundation	Lancome	81	531
24	1	Rouge Dior Ultra Lipstick	Dior	81	531
37	2	Ultra Long Wear Foundation	Lancome	87	531
38	1	Rouge Dior Ultra Lipstick	Dior	87	531
36	1	Soy Face Cleanser	Fresh	87	531
39	20	Soy Face Cleanser	Fresh	88	531
40	2	Ultra Long Wear Foundation	Lancome	88	531
41	1	Rouge Dior Ultra Lipstick	Dior	88	531

(19 rows)

pid	cid	trackingnumber	pdate	orderdetail
-----	-----	----------------	-------	-------------

88	531		2019-03-24 00:17:03.928544	Ordered Items: Soy Face Cleanser Fresh * 20, Ultra Long Wear Foundation Lancome * 2, Rouge Dior Ultra Lipstick Dior * 1, Total cost: 142
----	-----	--	----------------------------	--

(1 row)

2. Application

1. Admin Functions

- 1.1. Login using the Administrator's password, in our system, we set GLLZ as the Password, and by entering the wrong password, the system will remind you to enter the right one.

```
-----Welcome to Sephora Online System-----
Please select your role:
1 - Administrator
2 - Customer
1
Please enter the Administrator Password:
password
You entered the wrong password. And please do again!
Please enter the Administrator Password:
GLLZ
You successfully logged in!
```

After the Administrator logging in the system, the application provides 3 options to select with:

```
Dear Administor, welcome back to our system, please select the option you want to make:
1 - Search customer's information
2 - Change stock
3 - Operate the cart items
0 - Quit the application
Please type the number of the option :
1
```

1.2. Search customer's information

Our system provides three ways to search customer's information: With first name, last name or full name.

```
Please select the options you want to search:
1 - By Full Name.
2 - By First Name.
3 - By Last Name.
```

1.2.1 Search customer's information with his/her full name:

```
Please select the options you want to search:
1 - By Full Name.
2 - By First Name.
3 - By Last Name.
1
Please enter the full name of user:
Bessie Luo
Here is the information you want to find from this name:
cid          email          birthday          fullname
-----
534          bessie.luo@gmail.com    1998-01-26        Bessie Luo
```

1.2.2 Search customer's information with his/her first name:

```
Please select the options you want to search:
1 - By Full Name.
2 - By First Name.
3 - By Last Name.
2
Please enter the first name of user:
Bessie
Here is the information you want to find from this name:
cid      email      birthday      fullname
-----
534      bessie.luo@gmail.com      1998-01-26      Bessie Luo
```

1.2.3 Search customer's information with his/her last name:

```
Please select the options you want to search:
1 - By Full Name.
2 - By First Name.
3 - By Last Name.
3
Please enter the last name of user:
Luo
Here is the information you want to find from this name:
cid      email      birthday      fullname
-----
534      bessie.luo@gmail.com      1998-01-26      Bessie Luo
```

1.3 Change stock:

Administrator can get the detail about the out of stock products details, this means the product whose ordered quantity is greater than its stock.

```
Dear Administor, welcome back to our system, please select the option you want to make:
1 - Search customer's information
2 - Change stock
3 - Operate the cart items
0 - Quit the application
Please type the number of the option :
2
Here is all the details about out of stock products:
pname      ordered quantity      stock
-----
Refreshing Eye Cream      10      6
Soy Face Cleanser      24      20
```

1.3.1 Then administrator can do operations on the stock quantity by increase or decrease the number of stock.

```
Here is all the details about out of stock products:
pname      ordered quantity      stock
-----
Refreshing Eye Cream      10      6
Soy Face Cleanser      24      20
Please enter the name of the product that you want to change stock:
Soy Face Cleanser
Please enter the option that you want to do with the stock:
1 - Increase
2 - Decrease
1
Please enter the quantity you want to change with the stock:
10
You successfully changed stock!
```

- 1.4 Operate the cart items. This section, the administrator just delete entries from CartItems after customers checks out because now these items have been added to OrderItems.

```
Dear Administor, welcome back to our system, please select the option you want to make:
1 - Search customer's information
2 - Change stock
3 - Operate the cart items
0 - Quit the application
Please type the number of the option :
3
You successfully delete all entries from cartitem after customer checks out!
```

- 1.5 Exit the system

```
Dear Administor, welcome back to our system, please select the option you want to make:
1 - Search customer's information
2 - Change stock
3 - Operate the cart items
0 - Quit the application
Please type the number of the option :
0
Have a nice day :)
Thank you for using our online system!
```

2. Customer Functions

- 2.1. Login using his/her email

```
-----Welcome to Sephora Online System-----
Please select your role:
1 - Administrator
2 - Customer
2
Please enter your user email:
bessie.luo@hotmail.com
Your user email does not exist please try again.
Please enter your user email:
bessie.luo@gmail.com
You successfully logged in!
```

After the customer log in, the application provides with 5 options to select with

```
----- Sephora -----
1 - Search Product
2 - Most popular Product
3 - Ordered Items
4 - Cart Items
5 - Rate
6 - Submit Payment
0 - Quit the application
Please type the number of the option :
1
```


2.2. Search products

2.2.1. By keyword in brand name

```
Please type the number of the option :
1
Search by:
1. Brand
2. Category
3. Product name
1
Enter Brand name
Fresh
```

Product	Category	Brand
Soy Face Cleanser	Facial Cleanser	Fresh
Lotus Youth Preserve Moisturizer	Moisturizer	Fresh
Night Cream	Lotion	Fresh

2.2.2. By category

```
Search by:
1. Brand
2. Category
3. Product name
2
Enter Category name
Lipstick
```

Product	Category	Brand
Rouge Dior Ultra Lipstick	Lipstick	Dior
Powder Kiss Lipstick	Lipstick	MAC
Lip Color	Lipstick	TOM FORD

2.2.3. By keyword in product name

```
Search by:
1. Brand
2. Category
3. Product name
3
Enter Product name
Lip
```

Product	Category	Brand
Rouge Dior Ultra Lipstick	Lipstick	Dior
Powder Kiss Lipstick	Lipstick	MAC
Lip Color	Lipstick	TOM FORD

2.3 Get the most popular product in our store (popularity determined by the number of sells)

```
Please type the number of the option :
2
```

Product	#Sells	Rating
Soy Face Cleanser	24	1

2.4 List the items you have ordered by payment id

Please type the number of the option :

3

Product	Brand	Qty	PID
Rouge Dior Ultra Lipstick	Dior	1	71
Soy Face Cleanser	Fresh	1	71
Ultra Long Wear Foundation	Lancome	2	71
Rouge Dior Ultra Lipstick	Dior	1	72
Soy Face Cleanser	Fresh	3	72
Ultra Long Wear Foundation	Lancome	2	72
Rouge Dior Ultra Lipstick	Dior	1	73
Soy Face Cleanser	Fresh	3	73
Ultra Long Wear Foundation	Lancome	2	73

2.5 Cart Items

Please type the number of the option :

4

1. View cart
2. Add to cart
3. Delete from cart
4. Return to Main menu

2.5.1 View your shopping cart

1. View cart
2. Add to cart
3. Delete from cart
4. Return to Main menu

1

Item ID	Product	Brand	Qty
22	Night Cream	Fresh	1

2.5.1 Add items to cart (Qty cannot be ≤ 0)

Enter the product: *Soy Face Cleanser*
Enter the Brand: *Fresh*
Enter the quantity: *0*
Qty cannot be less than 0

Enter the product: *Soy Face Cleanser*
Enter the Brand: *Fresh*
Enter the quantity: *3*

1. View cart
2. Add to cart
3. Delete from cart
4. Return to Main menu

1

Item ID	Product	Brand	Qty
22	Night Cream	Fresh	1
23	Soy Face Cleanser	Fresh	3

2.5.1 Delete items from cart

Cannot delete if qty to delete exceeds the current quantity

```
Enter the Item ID: 23
Enter the qty to delete: 2
cannot delete more than the current qty
```

```
3
Enter the Item ID: 23
Enter the qty to delete: 2

1. View cart
2. Add to cart
3. Delete from cart
4. Return to Main menu
1
```

Item ID	Product	Brand	Qty
22	Night Cream	Fresh	1
23	Soy Face Cleanser	Fresh	1

If qty to delete equals the current quantity, then the item will be deleted from the cart

```
Enter the Item ID: 23
Enter the qty to delete: 1

1. View cart
2. Add to cart
3. Delete from cart
4. Return to Main menu
1
```

Item ID	Product	Brand	Qty
22	Night Cream	Fresh	1

2.6 Rate

2.6.1 Add a rating

```
1. Add Rating
2. View My Ratings
1
Enter the product: Channel 5
Enter the Brand: Channel
Enter your rating: 4
Enter your comment: A very signature and unique perfume
```

2.6.2 View your ratings

```
1. Add Rating
2. View My Ratings
2
```

Product	Brand	Rate	Comment
Channel 5	Channel	4	A very signature and unique perfume

2.7 Submit payment

When customer pays for all of his current cart items (Night Cream). We can see after this operation is carried out, there is no longer any cart items left and the product is added to the order items.

```
----- Sephora -----
1 - Search Product
2 - Most popular Product
3 - Ordered Items
4 - Cart Items
5 - Rate
6 - Submit Payment
0 - Quit the application
Please type the number of the option :
6
All your cart items have now been paid
----- Sephora -----
1 - Search Product
2 - Most popular Product
3 - Ordered Items
4 - Cart Items
5 - Rate
6 - Submit Payment
0 - Quit the application
Please type the number of the option :
4
1. View cart
2. Add to cart
3. Delete from cart
4. Return to Main menu
1

-----
Item ID      Product                                Brand      Qty
-----
```

```
Please type the number of the option :
3

-----
Product                                Brand      Qty      PID
-----
Rouge Dior Ultra Lipstick              Dior        1         71
Soy Face Cleanser                       Fresh        1         71
Ultra Long Wear Foundation              Lancome     2         71
Rouge Dior Ultra Lipstick              Dior        1         72
Soy Face Cleanser                       Fresh        3         72
Ultra Long Wear Foundation              Lancome     2         72
Rouge Dior Ultra Lipstick              Dior        1         73
Soy Face Cleanser                       Fresh        3         73
Ultra Long Wear Foundation              Lancome     2         73
Night Cream                             Fresh        1         79
```

3. Index

(1)after creating the index, by taking the average run time we can see, the queries run faster.

```
create index pdate_index on payments(pdate);
cluster payments using pdate_index;
--an index on pdate is useful because it sorts the payments on one customer's purchases
--by the date it was created. This will help execute SELECT statements based on pdate of
--payments faster since the matching tuples for that SELECT statement will be clustered
--on a few data pages, it will do even better for statements that specify a range of pdate
```

(2)after creating the index, by taking the average run time we can see, the queries run faster.

```
select * from rate;
select * from rate where rating=5;
--average time before indexing:61+61+60+61+62/5
--average time after...:60+61+57+58+60/5 less than before

create index rating_index on rate(rating);
cluster rate using rating_index;
drop index rating_index;
--an index on rating is useful because it sorts the products by the ratings they achieved
--This will help execute SELECT statements based on a specific rating number faster
-- since the matching tuples for that SELECT statement will be clustered
--on a few data pages, it will do even better for statements that specify a range of ratings
--note: rating is not the primary key or unique constraint
```

4. Visualization

(1) select all products with bad comments, and draw a chart to visualize how many bad votes for each of these products. Such data perhaps could help the store decide to decrease the stock for these particular low-rating products and increase the stock for other products.

SQL:

```
SELECT pname, bname, count(*)
FROM rate
WHERE rating = 1
GROUP BY pname, bname;
```

--then use the "copy" command to load the table to excel file

AutoSave OFF BadFeedbackProduct - Excel

File Home Insert Draw Page Layout Formulas Data Review View Help Team Tell me what you want to do

Paste

Clipboard

Font

Alignment

Number

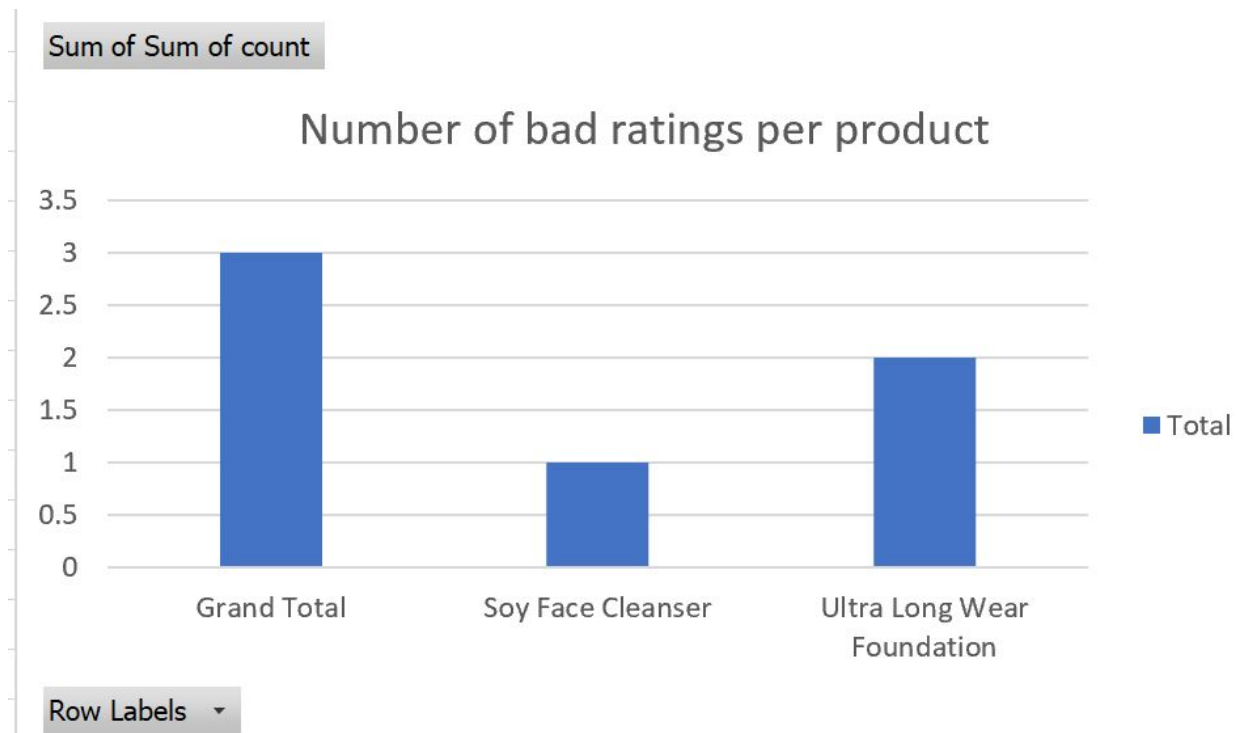
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel workbook.

A1 Row Labels

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Row Labels	Sum of count											
2	Soy Face Cl	1											
3	Ultra Long	2											
4	Grand Tota	3											
5													

--Here x-axis represents the pname, y-axis represents how many bad comments this product received

Row Labels	Sum of count
Soy Face Cleanser	1
Ultra Long Wear Foundation	2
Grand Total	3



(2) This chart visualizes how many orders have been made for each day in one week. The sales on weekdays are the lowest, and sales went up on Fridays and during the weekend. By observing these data, we could design sales strategies to introduce promotions during the weekdays to promote more users to shop online and new items on Fridays or weekends because that's when most users browse our site.

```
with order_by_day AS(
    select date_part('dow', pdate)
    from payments
    group by weekday
    order by weekday
)
select
case
    when weekday=0 then 'Sunday'
    when weekday=1 then 'Monday'
    when weekday=2 then 'Tuesday'
    when weekday=3 then 'Wednesday'
    when weekday=4 then 'Thursday'
    when weekday=5 then 'Friday'
    when weekday=6 then 'Saturday'
end as order_day, count
from order_by_day;
```

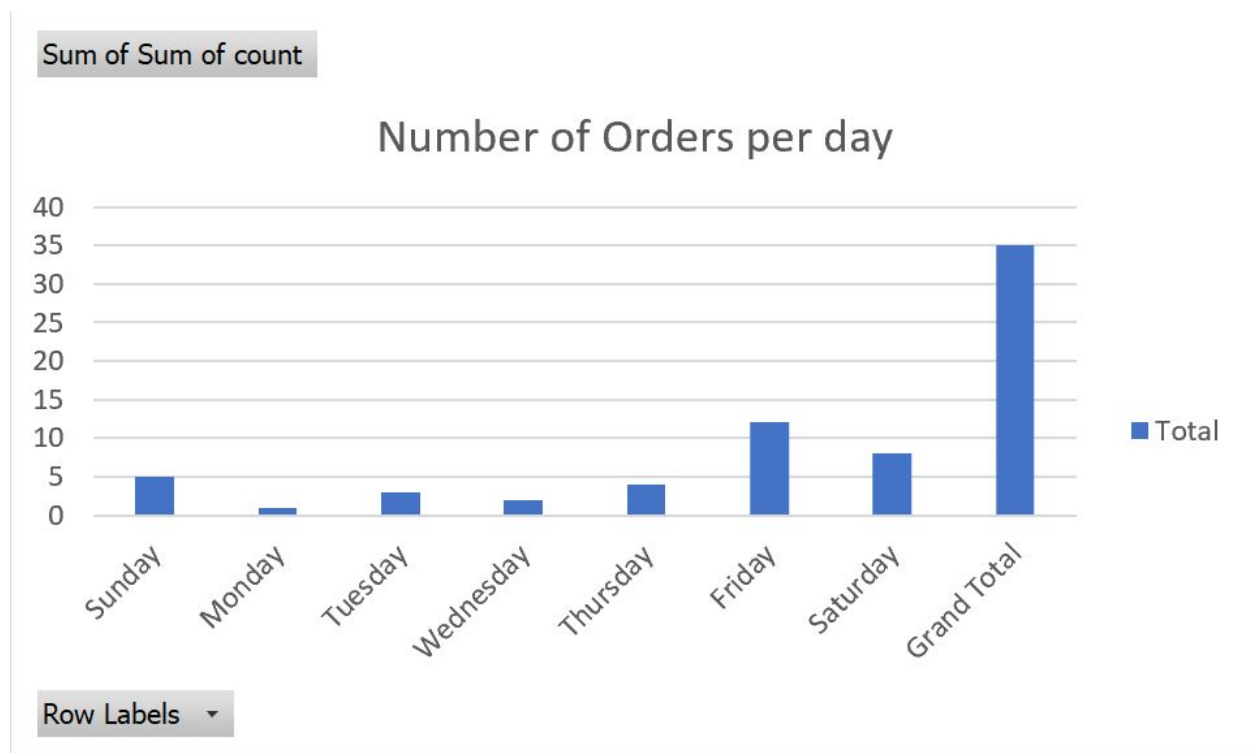
Data Output		Explain	Messages
	order_day text	count bigint	
1	Sunday	5	
2	Monday	1	
3	Tuesday	3	
4	Wednesday	2	
5	Thursday	4	
6	Friday	12	
7	Saturday	8	

--then use copy command to load the table to excel file

[illegible]

--here x-axis represents each weekday in one week, y-axis represents how many orders have been processed for each weekday.

Row Labels	Sum of count
Sunday	5
Monday	1
Tuesday	3
Wednesday	2
Thursday	4
Friday	12
Saturday	8
Grand Total	35



5. Creativity

For creativity part we did the following:

1. Visualization using Python

We wrote a script in python reading .csv file and plot it with Python matplotlib

```
#!/usr/bin/env python
import matplotlib.pyplot as plt
import csv

x = []
y = []

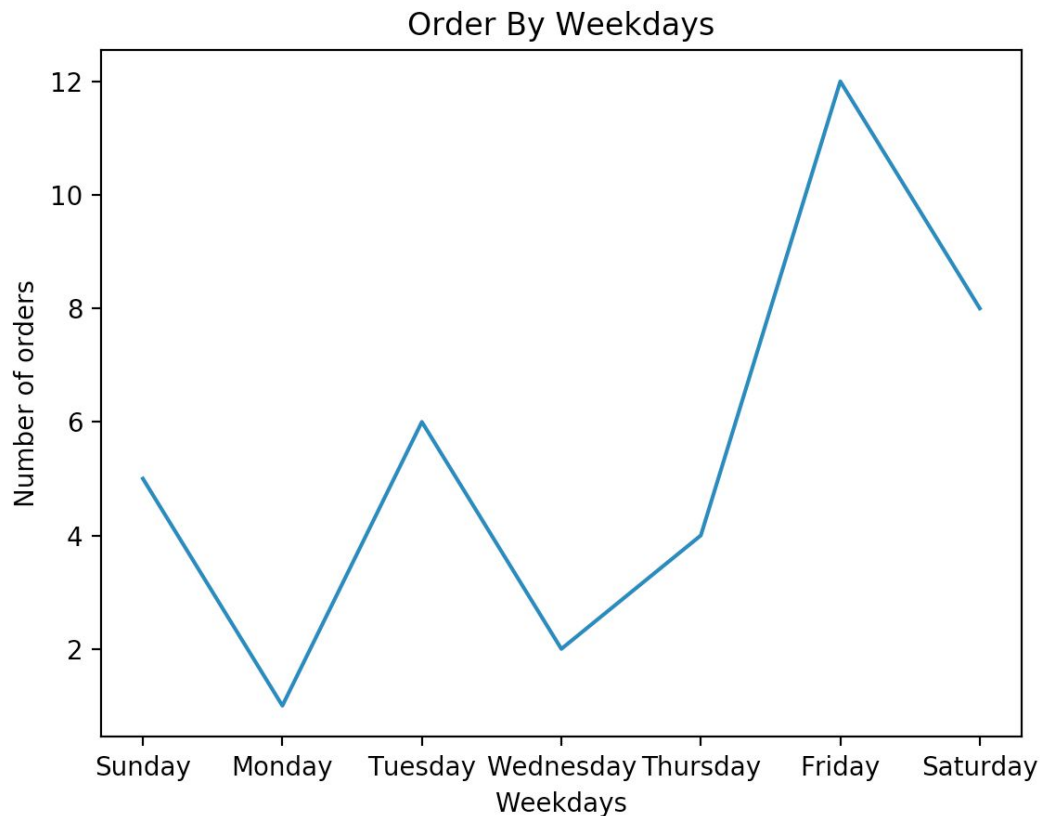
with open('orderByDay.csv', 'r') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(row[0])
        y.append(int(row[1]))

plt.plot(x, y)
plt.xlabel('Weekdays')
plt.ylabel('Number of orders')
plt.title('Order By Weekdays')
plt.show()
```

A brief look at the orderByDay.csv file

	A	B	C
1	Sunday	5	
2	Monday	1	
3	Tuesday	6	
4	Wednesday	2	
5	Thursday	4	
6	Friday	12	
7	Saturday	8	
8			
9			

And after loading the csv file, the plot looks like this. We can see that the number of sales went up rapidly on Fridays and Weekends and is lowest on Mondays.



2. One more executable procedure(function) for the database system

This function handles the refund operation, when a customer wants to refund a specific item that he/she bought in one specific payment, then it will first check if item exists in the payment history, matching with the correct payment id and customer id, then it will delete that item from OrderItems, as well as create a single record in Refund table indicating the refund operation is carried out successfully.

```
-- refund a item
-- delete the item from OrderItems, and generate a record in refund table
CREATE OR REPLACE FUNCTION refund(IN refund_cid INT, IN refund_pid INT, IN refund_pname VARCHAR(30),
                                  IN refund_bname VARCHAR(30), IN refund_quantity INT) RETURNS VARCHAR(30)
AS $$
DECLARE quan INT;
DECLARE msg TEXT;
BEGIN
IF EXISTS (SELECT quantity FROM OrderItems WHERE pname = refund_pname AND bname = refund_bname AND pid = refund_pid)
THEN
SELECT quantity INTO quan FROM OrderItems WHERE pname = refund_pname AND bname = refund_bname AND pid = refund_pid;
IF (quan = refund_quantity)
THEN
DELETE FROM OrderItems WHERE pname = refund_pname AND bname = refund_bname AND pid = refund_pid;
msg := 'Refund successfully';
ELSE
UPDATE OrderItems SET quantity = quan - refund_quantity WHERE pname = refund_pname AND bname = refund_bname AND pid = refund_pid;
msg := 'Refund successfully';
END IF;
-- insert a record into refund
INSERT INTO Refund (cid, pid) VALUES (refund_cid, refund_pid);
ELSE
msg := 'No record of this item or this pid was found';
END IF;
RETURN msg;
END
$$ LANGUAGE plpgsql;
```

Query statement before the stored procedure is executed:

```
[cs421=> SELECT * FROM refund WHERE cid = 599;
```

cid	pid	refund_date
-----	-----	-------------

-------	--	--

(0 rows)

```
[cs421=> SELECT * FROM orderitems WHERE cid = 599;
```

orderitem_id	quantity	pname	bname	pid	cid
--------------	----------	-------	-------	-----	-----

-------	--	--	--	--	--

9	2	Refreshing Eye Cream	Origins	36	599
---	---	----------------------	---------	----	-----

20	7	Refreshing Eye Cream	Origins	75	599
----	---	----------------------	---------	----	-----

(2 rows)

Query statement after the stored procedure is executed:

```
[cs421=> SELECT refund(599,75,'Refreshing Eye Cream','Origins',4);
```

```
refund
```


Refund successfully

(1 row)

```
[cs421=> SELECT * FROM refund WHERE cid = 599;
```

cid	pid	refund_date
-----	-----	-------------

-------	--	--

599	75	2019-03-24 00:50:57.606802
-----	----	----------------------------

(1 row)

```
[cs421=> SELECT * FROM orderitems WHERE cid = 599;
```

orderitem_id	quantity	pname	bname	pid	cid
--------------	----------	-------	-------	-----	-----

-------	--	--	--	--	--

9	2	Refreshing Eye Cream	Origins	36	599
---	---	----------------------	---------	----	-----

20	3	Refreshing Eye Cream	Origins	75	599
----	---	----------------------	---------	----	-----

(2 rows)

We can see that the quantity ordered has changed from 7 to 3