CSCI-3901-Assignment6-Problem2-Argument

Name:	Pratheep Kumar Manoharan
Banner Id:	B00837436
Files:	
	CSCI-3901-Assignment6-Problem1.pdf
	CSCI-3901-Assignment6-Problem2.pdf
	CSCI-3901-Assignment6-Problem2-Argument.pdf
	SQL_Statements_Problem2.sql

Objective:

To provide the necessary evidence that the program is working as expected.

Meeting the requirements:

All the deficiencies mentioned in the requirements are addressed. Please find the details of the remedy provided in the next section and how it is achieved. Each line from the requirement is converted into user stories. The test cases are created with the user stories.

Remedy provided:

1. *Modifying the database so that we can track orders from suppliers, know the cost at which we bought the product, know when the products have arrived, know who will be delivering the product to us, and have a reference to track the product while it is in transit.*

New table Purchases is created. Please find below the table explaining how the remedy is provided. Each line is taken as a user story and addressed directly.

S.No	Description	Column used	Column description
1	track orders from suppliers	purchaseid	Formed by combining purchase date and supplier id. Purchase Id and product id combination is unique for the given date.
2	know the cost at which we bought the product	unitprice	Stores the unit buying price of the product.
3	know when the products have arrived	receivedate	stores the product received date.
4	know who will be delivering the product to us	shipvia	stores the shipping company Id.
5	have a reference to track the product while it is in transit	purchaseid	Formed by combining purchase date and supplier id. Unique for the given date

2. *Record when an order for one of our own clients is shipped, update the inventory, and automatically trigger a reorder from the supplier, if necessary, knowing that the number to reorder can vary by product. We only want one reorder to be sent at the end of the day so we'll have two methods:*
    a. *Ship_order – happens for each order that we send out*
    b. *Issue_reorders – happens once per day where we place orders to our own suppliers; we only want at most one order to suppliers each day.*

    Please find below the remedy provided to address this requirement. The existing columns are reused to address the requirements.

S. No	**Description**	**Column used**	**Table**	**Description**
1	Record when an order for one of our own clients is shipped	Shippeddate	Orders	NA
2	update the inventory	unitsinstock, unitsonorder	Products	NA
3	automatically trigger a reorder from the supplier	NA	NA	When products is not in stock, issue reorder method is called to place reorder with current date. In case for the current date, the reorder is already for the supplier, then error message will be printed to user to place reorder manually.

3. *Record when we receive an order from one of our suppliers.*

    We will use the receivedate in purchases table to record the received date. Unitsinstock and unitsonorder columns of products table will be used to update the received inventory details.

Accuracy:

- All the requirements are addressed. The solution provided is tested thoroughly.
- All the attached test cases are executed in bluenose server and achieved to get the expected outcome.
- After each test case, the inventory is verified in the products table.
- Receive order method is used to verify the issue reorder orders. Ship order is used to ship the orders received. In this cycle for each product combination, I have covered all the cases of the entire program.

I hope I have covered all possible test cases. So, I believe the program operates as required.

Robustness:

All the exceptions are handled. The specific exceptions are caught, and the specific message is printed to the user. For the ship order and receive order methods, the program will throw order exception with error message and reference.

Test cases:

Please find below the set of test cases executed.

S.No	Flow	Test case	Expected result
Ship Order			
1	ship order	Give an invalid order id negative or 0.	Ship order should get failed printing the error message and reference.
2	ship order	Give an invalid order id which is not in database.	Ship order should get failed printing the error message and reference.
3	ship order	Give an order id which is already shipped.	Ship order should get failed printing the error message and reference.
4	ship order	Give a valid order id with only one product with enough inventory to ship.	Ship order should pass. Inventory should be updated correctly.
5	ship order	Give a valid order id with more than one product with sufficient inventory to ship.	Ship order should pass. Inventory should be updated correctly.
6	ship order	Give a valid order id with one product equal to inventory to ship.	Ship order should pass. Inventory should be updated correctly.
7	ship order	Give a valid order id with one product but no inventory to ship.	Ship order should get failed printing the error message and reference. First update should be rolled back in db.
8	ship order, issue reorder	Give a valid order id with one supplier one product. Inventory available should be less than reorder level. Unitsonorder should be 0 for that product.	Reorder should be placed. Inventory should be updated.
9	ship order, issue reorder	Give a valid order id with one supplier one product. Inventory should not be less than reorder level. Unitsonorder should be 0 for that product. Reorder level is zero or nonzero less than unitsinstock.	Reorder should be placed. Inventory should be updated.
10	ship order, issue reorder	Test for the same order Id. With unitsonorder waiting.	Reorder is already placed and waiting to receive the order. Ship order and issue reorder both should get failed.

11	ship order, issue reorder	Give a valid order id with one supplier more products. Inventory available should be less than reorder level. Unitsonorder should be 0 for that product.	Reorder should be placed. Inventory should be updated.
12	ship order, issue reorder	Give a valid order id with one supplier more products. Inventory should not be less than reorder level. Unitsonorder should be 0 for that product. Reorder level should be zero or less than unitsinstock. Quantity to ship should be more than unitsinstock.	Reorder should be placed. Inventory should be updated.
13	ship order, issue reorder	Test for the same order Id. With unitsonorder waiting. One supplier more product.	Reorder is already placed and waiting to receive the order. Ship order and issue reorder both should get failed.
14	ship order, issue reorder	Give a valid order id with more than one supplier with one product for each supplier. Inventory available should be less than reorder level. Unitsonorder should be 0 for that product.	Reorder should be placed. Inventory should be updated.
15	ship order, issue reorder	Give a valid order id with more than one supplier with more than one product for each supplier. Inventory available should be not less than reorder level. Unitsonorder should be 0 for that product.	Reorder should be placed. Inventory should be updated.
16	ship order, issue reorder	Test for the same order Id. With unitsonorder waiting. More than one supplier with one product each.	Reorder is already placed and waiting to receive the order. Ship order and issue reorder both should get failed.
17	ship order, issue reorder	Give a valid order id with more than one supplier with more than one product for each supplier. Inventory available should be less than reorder level. Unitsonorder should be 0 for that product.	Reorder should be placed. Inventory should be updated.
18	ship order, issue reorder	Test for the same order Id. Products ordered for the supplier.	Reorder is already placed and waiting to receive the order. Ship order and issue reorder both should get failed.
Issue Reorder			
19	issue reorder	Give negative value in year.	Method should return 0 and error message.
20	issue reorder	Give year as future value.	Method should return 0 and error message.

21	issue reorder	Give negative value in month.	Method should return 0 and error message.
22	issue reorder	Give value above 12 in month.	Method should return 0 and error message.
23	issue reorder	Give current year and month as future.	Method should return 0 and error message.
24	issue reorder	Give negative value in day.	Method should return 0 and error message.
25	issue reorder	Give value above 31 in day.	Method should return 0 and error message.
26	issue reorder	Give a future day.	Method should return 0 and error message.
27	issue reorder	give a invalid day like 30feb, 31Apr	Method should return 0 and error message.
28	issue reorder	Check leap year 29Feb2020	Reorder should be placed. Inventory should be updated.
29	issue reorder	Check non leap year 29Feb2019	Method should return 0 and error message.
30	issue reorder	Give a date for which reorder is already done for the supplier for the same product.	Method should return 0 and error message.
31	issue reorder	Give a date for which reorder is already done for the supplier but not for the same product.	Method should return 0 and error message.
32	issue reorder	Give a date for which reorder is already done but not for the supplier.	Reorder should be placed. Inventory should be updated.
33	issueReorder	Give a date when no products to reorder.	Method should return 0 and error message.
34	issueReorder	Test a valid scenario - one supplier one product.	Reorder should be placed. Inventory should be updated.
35	issueReorder	Test a valid scenario - one supplier many products.	Reorder should be placed. Inventory should be updated.
36	issueReorder	Test a valid scenario - more than one supplier with one product from each supplier.	Reorder should be placed. Inventory should be updated.
37	issueReorder	Test a valid scenario - more than one supplier with more than one product from each supplier.	Reorder should be placed. Inventory should be updated.
Receive Order			
38	receiveOrder	Give purchase reference for which order is already received.	Receive order should fail printing the message and reference.

39	receiveOrder	Give an invalid purchase id	Receive order should fail printing the message and reference.
40	receiveOrder	Test a valid scenario for the day with one supplier with one product reordered.	Receive order should pass. Inventories should be updated.
41	receiveOrder	Test a valid scenario for the day with one supplier with more than one product reordered.	Receive order should pass. Inventories should be updated.
42	receiveOrder	Test a valid scenario for the day with more than one supplier with one product from each supplier reordered.	Receive order should pass. Inventories should be updated.
43	receiveOrder	Test a valid scenario for the day with more than one supplier with more than one product from each supplier reordered.	Receive order should pass. Inventories should be updated.
44	receiveOrder	Receive order details updated in purchase, purposefully fail while updating inventory.	Receive order should fail printing the message and reference.