

Assignment – 6

Make some meaningful changes to a database (Problem-2)

Overview

The Northwind food distribution company wants to improve its inventory management system by modifying their database. It has found some deficiencies and requires to solve some problem which are it doesn't have history of purchases or the price at which it bought products, when it places an order to restock, the products are ordered manually and it doesn't know when the products arrive from the supplier.

So, to solve this issue I have created a program which will calculate the purchase history from when the company started till today's date. In this program a table will be created in the database, which will give information about when and how much units of a particular product are been reordered. Also, the program will calculate at what price the supplier had sold the products to the company on the day we ordered a product and what was the total bill of the order. This will help company to review how much quantity of products had been ordered and how frequently it was ordered

Next, I have created a separate interface class which implements three methods. First is Ship Order, this method will be called whenever an order of a client is shipped. Second method is Issue Reorder, this method will issue an order to the shipper when the products reaches below reorder level. Third method will notify the company as soon as the order from the supplier arrives from the company.

Working

Expected Purchase History Program

In this program, first of all DriverClass connects to the database of the company. Next, the query to retrieve data from the database necessary for the calculation of the expected purchase history will be executed. Once the query is executed all the necessary information required to calculate the expected purchase history are stored in variables. First of all, the Product Id will be checked and calculations will be done for every product Id separately. For every unique Product Id first of all the day end inventory for the last date on which company ordered from the supplier will be stored in the Day_End_Inventory variable. Next the createFirstPurchaseHistory() will be called of CreateHistoryClass. In this method we will check for three conditions. First condition is if there are UnitsOnOrder available for the product, reorder units will be considered equal to UnitsOnOrder. This calculation is only done for the very last entry of the product ordered to the supplier. After that the program will backtrack and calculate reorder units as DayEndInventory – Reorderlevel till a new unique Product ID is encountered. Second Condition is if the ReorderLevel and UnitsOnOrder both are 0 and DayEndInventory which is basically UnitsInStock from the database. In this condition we calculate reorderunit as DayEndInventory – SalesDayEndInventory which is basically the sales on a particular day. In third Condition createPurchaseHistory() method of the CreateHistoryClass will be called which have a condition that checks if DayEndInventory of is greater than or equal to reorder level * 4, the program will set Salesdayendinventory equal to ReorderLevel and reorderunit will be Day_End_Inventory – ReorderLevel. Every result calculated for the reoder unit will be append in the String Buffer. Likewise the whole expected purchase history table will be created . To calculate at what price the company bought the product, I have used a formula which is UnitPrice / 1.15 which will give the unit price at which products were bought. To calculate the total bill, I have multiplied reorderunits by unit price. Next the table named purchase history will be created and the whole purchase history data along with the unit price at which the product was bought will inserted into the table purchase history.

InventoryControl Interface

- 1) Ship_order(int orderNumber) - This method will be used to indicate that the order placed by the customer has been shipped. First of all the query will be executed to fetch data of the given order number provided in the argument of the method. It will retrieve all the necessary data that will be used to update the database. First it will check if the Units In Stock is greater than Quantity. If the condition is true the program will subtract the quantity ordered by the customer from the UnitsInStock. Next it will update UnitInStock in the product table and ShippedDate will also be updated in the orders table to the today's date.
- 2) Issue_reorders(int year, int month, int day) – This method will issue reorder for a particular day passed in the argument, if the Units In Stock is less than ReorderLevel. For every supplier the order will be sent at the end of every day in one order. Finally the method will return the number of supplier from whom we will place the order.
- 3) Receive_order(int internal_order_reference) – This method will update a table that I have created for future operation of the company. This method will update the table with a reference number which will identify that the order has been received.

Assumptions

I have assumed that:

- The SQL to modify the database will also be executed on the database
- Re-creating the history will always be done before calling methods from the inventoryControl class
- When recreating the history of orders to suppliers, the inventory will always have hit exactly the reorder level to trigger the reorder
- When create purchase history when there is UnitsOnOrder in products and reorder level is zero programs will consider unitonorder equal to reorder unit for the first counting of recorder unit in every product
- The database username, password and credentials need to be changed in expected history class and all the interface methods