Assignment 5

Purvisha Patel

(B00912611)

Subject:

Software Development Concepts

Professor:

Mike McAllister

Problem#2

Overview

Learning to do some meaningful modifications to a database.

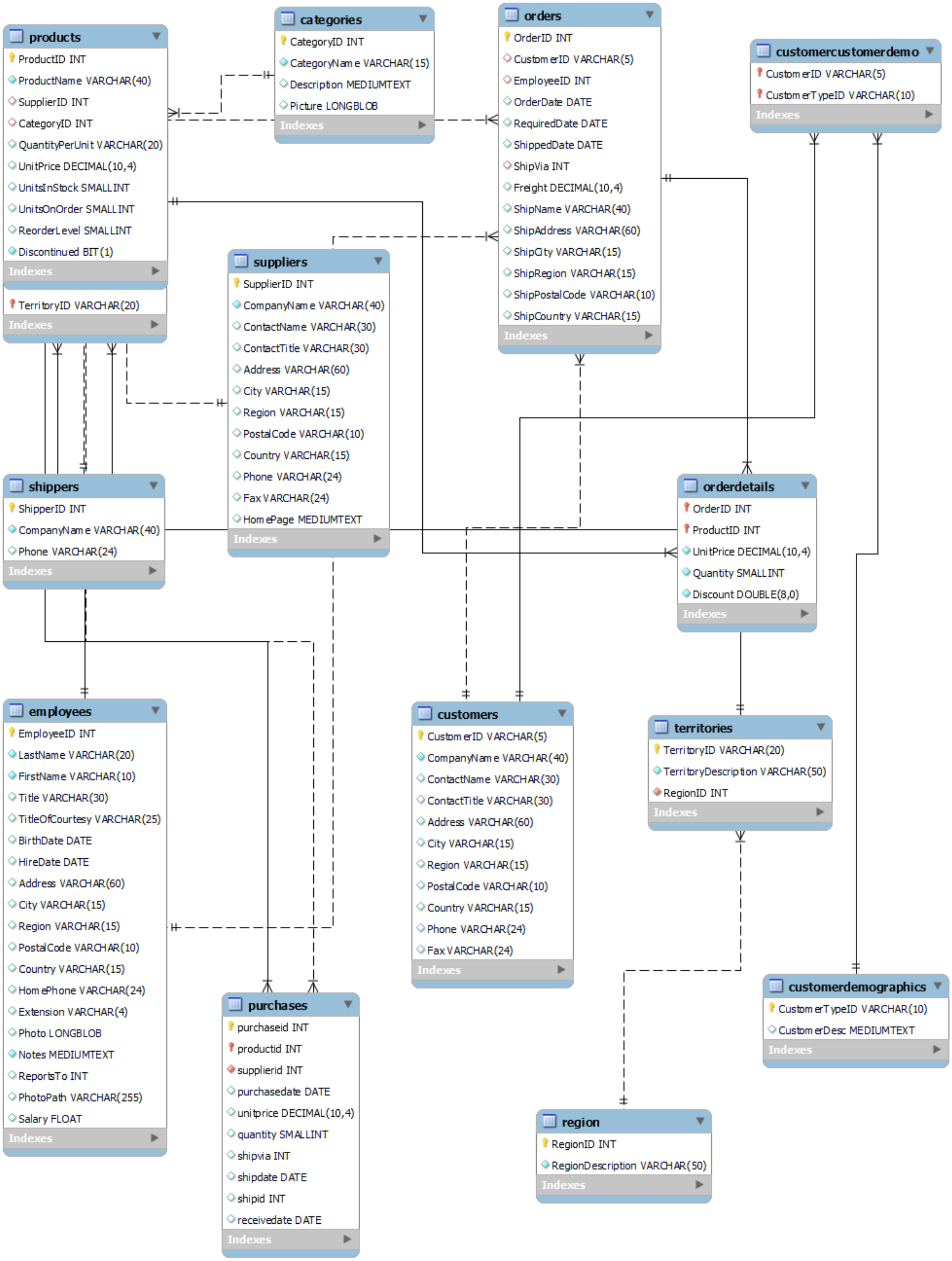
Assumptions

- On the same day, reorder cannot be placed with the same supplier. but can be placed with a different supplier.
- Reshipping an order that has already been shipped is not possible.
- Same order will be received just once.
- When u pass date as a parameter to the method of Issue_reorder it cannot be in the future, and it should be a valid date and not a null value or void.

Entity-relationship diagram

In the following ER diagram new table is added with the name "purchases".This table contains the following columns:

- purchaseid
- productid
- supplierid
- purchasedate
- unitprice
- quantity small
- shipvia
- shipdate
- shipid
- receivedate


products
ProductID INT
ProductName VARCHAR(40)
SupplierID INT
CategoryID INT
QuantityPerUnit VARCHAR(20)
UnitPrice DECIMAL(10,4)
UnitsInStock SMALLINT
UnitsOnOrder SMALLINT
ReorderLevel SMALLINT
Discontinued BIT(1)
Indexes
TerritoryID VARCHAR(20)
Indexes
categories
CategoryID INT
CategoryName VARCHAR(15)
Description MEDIUMTEXT
Picture LONGBLOB
Indexes
orders
OrderID INT
CustomerID VARCHAR(5)
EmployeeID INT
OrderDate DATE
RequiredDate DATE
ShippedDate DATE
ShipVia INT
Freight DECIMAL(10,4)
ShipName VARCHAR(40)
ShipAddress VARCHAR(60)
ShipCity VARCHAR(15)
ShipRegion VARCHAR(15)
ShipPostalCode VARCHAR(10)
ShipCountry VARCHAR(15)
Indexes
customercustomerdemo
CustomerID VARCHAR(5)
CustomerTypeID VARCHAR(10)
Indexes
suppliers
SupplierID INT
CompanyName VARCHAR(40)
ContactName VARCHAR(30)
ContactTitle VARCHAR(30)
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
Phone VARCHAR(24)
Fax VARCHAR(24)
HomePage MEDIUMTEXT
Indexes
shippers
ShipperID INT
CompanyName VARCHAR(40)
Phone VARCHAR(24)
Indexes
orderdetails
OrderID INT
ProductID INT
UnitPrice DECIMAL(10,4)
Quantity SMALLINT
Discount DOUBLE(8,0)
Indexes
employees
EmployeeID INT
LastName VARCHAR(20)
FirstName VARCHAR(10)
Title VARCHAR(30)
TitleOfCourtesy VARCHAR(25)
BirthDate DATE
HireDate DATE
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
HomePhone VARCHAR(24)
Extension VARCHAR(4)
Photo LONGBLOB
Notes MEDIUMTEXT
ReportsTo INT
PhotoPath VARCHAR(255)
Salary FLOAT
Indexes
customers
CustomerID VARCHAR(5)
CompanyName VARCHAR(40)
ContactName VARCHAR(30)
ContactTitle VARCHAR(30)
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
Phone VARCHAR(24)
Fax VARCHAR(24)
Indexes
territories
TerritoryID VARCHAR(20)
TerritoryDescription VARCHAR(50)
RegionID INT
Indexes
purchases
purchaseid INT
productid INT
supplierid INT
purchasedate DATE
unitprice DECIMAL(10,4)
quantity SMALLINT
shipvia INT
shipdate DATE
shipid INT
receivedate DATE
Indexes
customerdemographics
CustomerTypeID VARCHAR(10)
CustomerDesc MEDIUMTEXT
Indexes
region
RegionID INT
RegionDescription VARCHAR(50)
Indexes

Sequence of operations

Ship order

Conforming whether the order has been shipped or not.

- If this is not the case, an error message will be displayed.
- If yes, then the date of shipment will be replaced with the current date.

Determine whether there is sufficient stock to serve.

- If not, then we call issue reorder to place a new order with the present date.
- If yes, then we are prepared to ship.

In the product table, modify the inventory by deducting the shipped quantity from the column of units in stock.

When inventory runs out of stock, we calculate the reorder quantity by adding the order's quantity difference and reorder level.

Issue reorder

Determine whether we need to reorder any items.

- If there are no products to reorder, the execution will be terminated.
- If yes then, we will retrieve the information from the database.

Check to see if we have previously placed a reorder on the same date with the same suppliers.

- If already placed the reorder, then, we're done.
- If the reorder is left to be placed, proceed with it.

Make a new table and enter the details of reorder.
in the products table, Modify the reorder inventory, and in the unitsonorder column enter the quantity of reorder.

Receive order

Determine whether or not the order has been received.

- If it is received, the execution is terminated.
- If not, continue with the previously received order.

In the products table, modify the inventory of the received order by adding and deducting the quantity received in the unitsinstock column and quantity received in the unitsonorder column respectively.

Files and external data

Java Files

1. **DbConnection.java**

   This class is made to establish a connection with the database. It contains the username, password and database name which helps to connect with the database and also fetches the username, password and database name from the login.properties file.

2. **inventoryControl.java(interface)**

   This is the interface which contains definition of three methods

   - Ship_Order
   - Issue_reorders
   - Receive_order

3. **inventoryControlManagement.java**

   This is the main class that connects the main class with the rest of the program.

4. **login.properties**

   It contains username, password and database name.

5. **Main.java**

   This is the main class of the program. We can also call it as driver class of the Program

6. **OrderException.java**

   This is the user defined / User Created Exception that is used to print the customer messages to the users

7. **PurchaseOrder.java**

   This is the purchase order class that is used to set and get the values.

Remedies for the given Problems

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.

To overcome this situation I have created another table with the name “purchases” with the following columns:

- purchaseid : For tracking orders from the supplier while it is in transit
- unitprice : Cost of the product purchased by us
- shipvia : Information about the delivery
- receivedate : Arrival date of the products

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.

To overcome this I have used the existing columns mentioned below.

- Shippeddate: This will record when the client order is shipped.
- Unitsinstock and Unitsonorder: This will update the inventory

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

From the purchases table, the receivedate column will be storing the data of the received dates. From the product table, Unitsinstock and Unitsonorder columns will be storing the data of the updation of the Inventory.