SQL project with nice queries

Genera view:

The system create for mange warehouses and product at store.
The system contains members who can purchases products.
All the product organizes in categories and the system will create bills for every purchase.
If new product add to warehouse and warehouse's capacity is overflow the system will move the product for new free available warehouse.
The system contains manager that can appoint member to manager.
The system use triggers and procedures.

Tables description:

Warehouse:

- Unique id (PM)
- Unique name
- Maximum products capacity (no less the 20)
- Current capacity of product- cannot be less the 0 and more the maximum capacity

Category:

- Unique id (PM)
- Unique name – most be 3 or more characters.
- Total product in category

Product:

- Unique id (PM)
- Unique name
- Category name (FK)
- Warehouse name (FK)
- Entry date
- Price per unit – no less the 0 no more then 9999 with two numbers after the dot
- Maximum amount – no less then 20
- Current amount – no more the Maximum amount no less then 0
- Sell amount – the number of sold units

Users:

- Unique id (PK)
- Unique user name
- Password – most be between 8-20 characters
- Full name – most be 3 or more characters
- Personal unique id (PK)
- Date of birth – users must be at last 15 years old
- Phone number- must be 10 numbers, start with 05
- Gender – as m/f
- Registration date

- Is admin – 0/1

Invoices:

- Unique id (PK)
- User name (FK)
- User id – (FK)
- Date of purchase
- Total amount of product
- Total cost – default as 0 and update by trigger

Purchases:

- Unique id (PK)
- Invoice id (FK)
- Product id (FK)
- Category Name (FK)
- Total units
- Total cost

Procedures:

1. addNewUser:
   prucedure get user name, password, full name, personal id, date of birth, phone number, gender, and insert new user into users table
2. addNewProduct:
   prucedure get product name, category name, cost per unit, maximum units to instore (amount 20 units as default unless inserted different), current amount (0 as default unless inserted different)
3. addNewPurchase:
   procedure get product name, category name, num of units of the same product, costumer name, invoice id.
   If invoice id didn't inserted (new order) the procedure will create new empty invoice with user name and user id and return the new invoice id.
4. makeDiscount:
   procedure get discount precent, total amount at warehouse, user name, number of days.
   The procedure sand user name and number of days to returnUserProductsByDate function and for each value it return the procedure will make discount by discount precent it get in term that product current capacity is larger from total amount the procedure get as parameter.
   The idea is to get rid of product with big amounts.
5. showProductByDate:
   procedure get date and create temporary table of date from the given date till current date. After create temporary table the procedure will combine it within products table and present product name, category name, warehouse name, total available amount, sorted by available amount
6. userGraphPie:

procedure get user name and return data for graph pie, by category name and purchases present.

7. top10:
   procedure get category name and return details of 10 invoices which cost the most. Return data will organized by user name, date of purchase, cost, given category
8. updateProductAmount:
   procedure get product id, product amount. The procedure check if the amount to update is not overstep from maximum capacity of product.
   If the amount which given is over step of warehouse maximum capacity the procedure will move the product to another warehouse and show massage and update both warehouses current capacity, update current capacity of product and show massage for each process
9. topSale:
   procedure select the 10 best seller products and create discount by:
   - if product cost between 0-200, 15% discount
   - if product cost between 200-500, 10% discount
   - if product cost more than 500, 5% discount

   The return data present as list of products by id, name, total sold units, new price
10. endOfDay:
    procedure create end of day report and return data as product name, id, invoice id, category name, total sold units, total cost.
    Return as well the total cost of all purchases at that day.

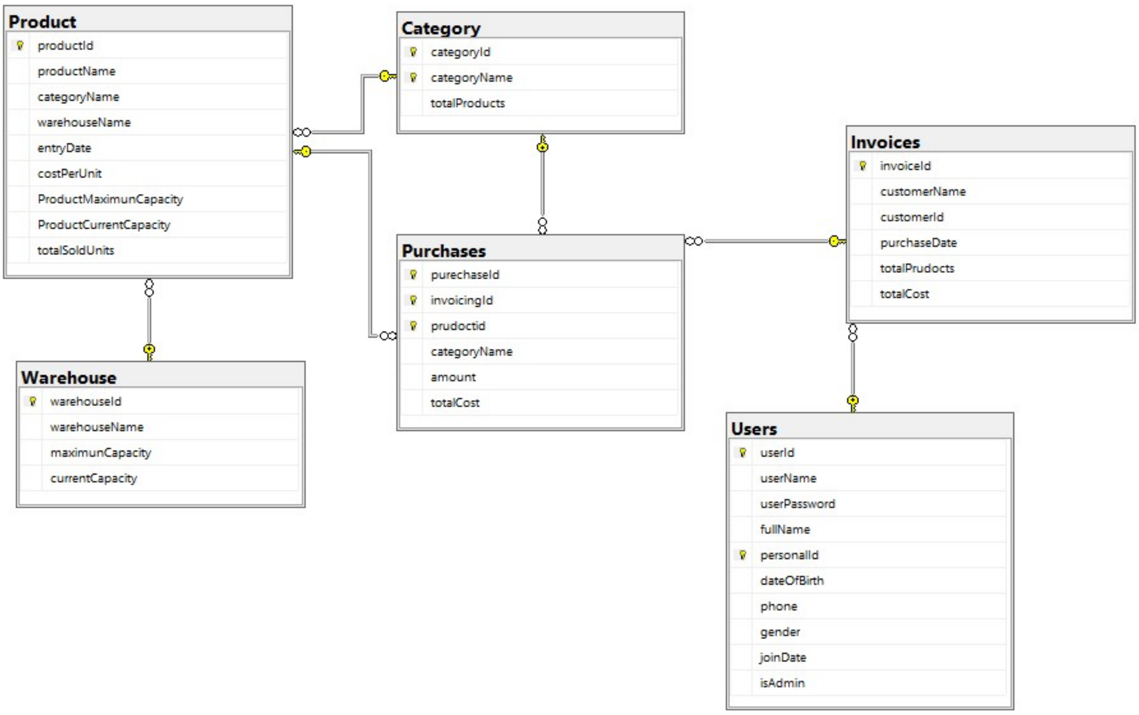
Triggers:

1. updatesForNewProduct:
   trigger for add new product will update:
   - number of products in category
   - will check if the product amount doesn't overstep warehouse maximum capacity, if it is will check for available warehouse and update both warehouses. If there is no available warehouse return callback with error massage
2. updateInvoice:
   trigger will update by invoice id the number of products, total cost of new porches. It will update as well the current amount of product, and warehouse current capacity

function:

1. returnUserProductsByDate:
   function get number of days and user name and return all the products he have buy from today and backward

Diagram:



5. פונקציה :

הסבר: פו נקציה המקבלת שם משתמש ו מספר ימים ומחזירה את כל המוצרים שהלקוח רכש במהלך כל הימים האלו.

קוד:

```
--5--
הפונקציה מקבלת  מספר ימים ושם משתמש  ומחזירה  את כל  המוצרים שהמשתמש רכש
-- בימים  אלו
create function returnUserProductsByDate(@userName nvarchar (40),
@numOfDays int) returns table as return
		(select * from Product as P where P.productId in(
						select prudoctid from Purchases where invoicingId in
(
								select invoiceId from
Invoices as I where I.customerName = @userName and
datediff(day,purchaseDate,getdate()) <= @numOfDays)))
```