**Department of Computer Science**

**The College at Brockport**

**CSC 423: Web Application Development**

**TERM PROJECT DESCRIPTION**

**(Obtained with the co-operation of the personnel of Aptaris, LLC)**

Our client, Josbrian, is a developer of web-based software for the retail industry. Josbrian wishes to develop software that will make it easier to generate the inserts announcing sales at various stores in your area that appear in your Sunday newspaper, get delivered to your mailbox, etc. Each such insert is called an “Advertising Event” in the industry, or “Ad Event” for short. Every “Ad Event” is tracked by Josbrian with a unique code, and is given a name as well (e.g., “Wegman’s Grand Sale”). Each such “Ad Event” will run for a definite time period – therefore, a start date and end date will be associated with it. Finally, for explanatory purposes, the “Ad Event” will also have a description associated with it, plus a “type” value that is used for internal tracking purposes.

An “Ad Event” could announce numerous different kinds of sales. For example, a particular “Ad Event” for Wegman’s could announce that the item “Atlantic Salmon” is going to be on sale for $ 2 off per pound, and it could also announce that “Kit Kat Candy” is on sale for 50% off per bag (say, after Halloween). Each such announcement is called a “Promotion” in industry jargon – each “Promotion” is also tracked by Josbrian with a unique code, is assigned a name, and has a description. But most importantly, each “Promotion” has two important data items – one called “Amount Off” and the other “Type”. “Amount Off” will take a numerical value, and “Type” can take one of two values – “Dollar” or “Percent”. Therefore, if in a certain “Promotion”, “Amount Off” has value of 25 and “Type” has the value “Percent”, then that promotion is announcing a sale of 25% on some item(s). Similarly, if some other “Promotion” has an “Amount Off” has value of 2 and “Type” has the value “Dollar”, then that promotion is announcing a sale of $2 on some item(s).

Finally, Josbrian needs to keep track of “Items” in its data repository. Each such item refers to entities like “Atlantic Salmon”, “Bananas”, etc. Each item has a unique item number, a description, a category and department (the store’s department it is stocked in – like “Produce” or “Seasonal”). More importantly, with each item, Josbrian keeps track of its full retail price and cost retailer incurred in purchasing the item.

Note that when Josbrian processes a sale announcement, it creates a number of “Promotion” objects. It then associates a number of “Item” objects with that “Promotion”. For example, if “Promotion” object ‘p’ is a $ 2 off sale, then all items that are going to be on sale for $ 2 off will be added to it – say, “Atlantic Salmon”, “Extra large shrimp”, etc. **NOTE:** The moment an “Item” is associated with a “Promotion”, its *sale price* must be immediately computed and stored in the data repository. For example, if the salmon’s full retail price was $ 13.99/lb, the sale price (if it gets associated with the $ 2 off promotion) will be immediately computed as $ 11.99/lb and stored. Note also that Josbrian reserves the right to go and change the details of the promotion. For example, they may decide that the $ 2 off promotion is actually going to be a $ 1.50 off promotion. If this is done, then the sale price of salmon should immediately be recomputed as $ 12.49/lb.

Several such “Promotion” objects could then be associated with a newly created “Ad Event”. Once an “Ad Event” runs its course, a new “Ad Event” could be created with new, or already existing, “Promotion” objects.

The functionality your system must provide includes the following:

* Add a new Item to the system
* Update details of an existing Item in the system
* Add a new Promotion to the system
* Search for an item (search can be by category/department, item number and/or description) and add it to a promotion (which may be searched for by promotion code, name and/or description). When an item is added to a promotion, its promotion retail price should be automatically calculated and stored for the item.
* Update a Promotion: this will allow the description, etc. of the promotion to be changed. But, primarily, this will change the sale details of the promotion (“Amount Off” and “Type”). When this happens, the promotion retail price for the item should be automatically updated.
* Add a new Ad Event to the system
* Search for a promotion (using promotion code, name and/or description) and add it to an Ad/Event (which may be search for by its unique code, date range, name and/or description).
* Update details of an Ad Event in the system.

Finally, Josbrian would like a number of reports. These include:

* Search for all events by date and report all the promotions associated with it.
* Search for all promotions associated with a certain amount off and type value.
* Tell me which ad/event gives me the biggest savings on an item? Savings is the difference between full retail price and promotion price. An item is identified by its item number, or after selecting an item searched for by description, category and/or department.
* Search for top 50 items on sale and tell me which events they are a part of. Again, here the savings is the difference between full retail price and promotion price.

**NOTE:** Proper GUI design is important in projects like these. Therefore, it is important that you make the GUI as user-friendly as possible and less prone to errors on the part of the user. How can this be done? For example, you should choose to provide the user with a drop-down list in the GUI instead of text fields when appropriate. Take a look at the attached ‘Item Master.xlsx’ Excel spreadsheet. This shows the data associated with the various items that will go into your Item table. Note that there is a Category and Department Name associated with each item. There is a pre-defined set of Category names and Department Names associated with the business – you can see the lists of names (at least, you can CREATE these lists of names) from the data in the spreadsheet. Therefore, when the GUI asks the user to provide a Category and/or Department Name, the user should not be allowed to TYPE THE NAME in into a text field – instead, it is important to allow the user to SELECT the name from a drop-down in the GUI. If you would like to add to the data model shown below and have a table for each of Category and Department Names – meaning, the entire set of possible Category names and the entire set of possible Department Names are in these tables, with each name in each table being unique – feel free to do that. At this point, the customer is not asking for names in these tables to be managed (i.e., no add/modify/delete on these names – you can create and populate these tables using phpmyadmin once and for all), but feel free to put in functionality to manage these tables if you have the time.

**DATABASE SCHEMA**

**Table Name: Item**

|  |  |
| --- | --- |
| ItemNumber | Text (unique, PK) |
| Item Description | Text |
| Category | Text (e.g. Seafood) |
| DepartmentName | Text (e.g. Produce, Seasonal) |
| PurchaseCost | Number |
| FullRetailPrice | Number |

**Table Name: Promotion**

|  |  |
| --- | --- |
| PromoCode | Text (unique, PK) |
| Name | Text |
| Description | Text |
| AmountOff | Number |
| PromoType | Text (percent, dollar) |

**Table Name: AdEvent**

|  |  |
| --- | --- |
| EventCode | Text (unique, PK) |
| Name | Text |
| StartDate | Text |
| EndDate | Text |
| Description | Text |
| Type | Text |

**Table Name: Promotion-Item (many-many relationship, as per problem description)**

|  |  |
| --- | --- |
| ID | AutoNumber |
| PromoCode | Text (FK to Promotion) |
| ItemNumber | Text (FK to Item) |
| SalePrice | Number (calculated using the PromoType and AmountOff of the associated Promotion) |

**Table Name: AdEvent-Promotion (many-many relationship, as per problem description)**

|  |  |
| --- | --- |
| ID | AutoNumber |
| EventCode | Text (FK to AdEvent) |
| PromoCode | Text (FK to Promotion) |
| Notes | Text |

**DATA MODEL**

