Triggers:

* all the added business functionality is encompassed in 1 apex Trigger per SObject.

How to temporary disable the triggers' functionality:

Disabling the any of the following trigger functionality can be done by deleting the related list values from the TriggerControl custom settings:

This example is for the MasterAccountTrigger:

* AccountTrigger : - disables the entire Account trigger
* AccountTriggerBeforeInsert : - disables the before Account insert events related functionality
* AccountTriggerBeforeUpdate: - disables the before Account update events related functionality
* AccountTriggerAfterInsert : - disables the after Account insert events related functionality
* AccountTriggerAfterUpdate : - disables the after Account after events related functionality
* AccountTriggerAfterDelete : - disables the after Account delete events related functionality
* AccountTriggerAfterUndelete : - disables the after Account undelete events related functionality

1. MasterAccountTrigger

**Description:**

Apex trigger for the Account object. This Trigger is using a helper class called ClassAccountHelper.

**Business Functionality:**

On before insert and before update events is calling the AccountQualityScore method from the ClassAccountHandler apex class which calculates a general score of the account based on the amount of information on the object.

On before insert events is calling the newAccountsDuplicateCheck method from the ClassAccountHandler apex class that for all non-admin users checks if the new added accounts are duplicates or not.

**Affected Salesforce Fields:**

Company\_Size\_\_c, Primary\_Sector\_\_c, Country\_\_c , Account\_Quality\_Score\_\_c

2. MasterOpportunityTrigger

**Description:**

Apex trigger for the Opportunity object. This Trigger is using a helper class called ClassOpportunityHandler.

**Business Functionality:**

On before update and insert events events is calling the SetSalesManager method from the ClassOpportunityHandler apex class which sets the opportunity owner field to the Delevery\_Owner\_\_c field

On  before update and insert events is calling the changeOppStatus method from the ClassOpportunityHandler apex class  that changes the Opportunity StageName field according to the Project\_Stage\_\_c values.

On before update event is calling the updateOpportunityCloseDate() method from the ClassOpportunityHandler apex class which updates the CloseDate of a opportunity to 7 days after the current date if the StageName is either 'In Progress' , 'Pending Payment', 'Complete' , 'Lost' or 'Rejected;

On after update,insert,delete and undelete events is calling the OppCountOnContact method which counts the open, closed and created opportunities with regards to the related contacts.

**Affected Salesforce Fields:**

OwnerId,StageName, CloseDate, Open\_Opp\_Count\_\_c , Closed\_Opp\_Count\_\_c, Created\_Opp\_Count\_\_c

3. MasterLeadTrigger

**Description:**

Apex trigger for the Leadt object. This Trigger is using a helper class called ClassLeadtHandler.

**Business Functionality:**

On before update and insert events is calling the LeadQualityScore()  method from the ClassLeadHandler apex class that calculates a custom lead score based on the amount of filed information.

On before insert and update events is calling the SuggestedEmail() method from the ClassLeadHandler apex class. [depreciated method].

On after insert, update, delete and undelete events is calling the LeadsAtAccount() method from the ClassLeadHandler that calculates the amount of leads related to a Account.

On after insert events is calling the LinkLeadToAccount() method from the ClassLeadHandler that sets the Account\_\_c field based on the Company\_\_c field.

On after insert events is calling the LeadGenQualifiedLeads() method from the ClassLeadHandler that sets the Lead status to 'Qualified Lead' for all the ones that a created from a 'Lead Gen' user profile.

**Affected Salesforce Fields:**

Suggested\_Email\_\_c, Lead\_Count\_\_c, Account\_\_c, Status

4. MasterContactTrigger

**Description:**

Apex trigger for the Contact object. This Trigger is using a helper class called ClassContactHandler

**Business Functionality:**

On before update and insert events is calling the ContactQualityScore  method from the ClassContactHandler apex class that calculates a custom contact score based on the amount of filed information.

On before update and insert events is calling the ContactEmailFormat  method from the ClassContactHandler apex class. [ depreciated method ]

On before update and insert , after delete and undelete events is calling the AccountEmailFormat  method from the ClassContactHandler apex class. [ depreciated method ]

On after insert, update, delete and undelete events is calling the ContactsAtAccount method from the ClassContactHandler that calculates the amount of contacts, contacts that created opportunities and contacts that closed opportunities related to a Account.

**Affected Salesforce Fields:**

 Contact\_Quality\_Score, Account.Contact\_CreatedOpp\_Count\_\_c, Account.Contact\_ClosedOpp\_Count\_\_c, Account.Contact\_Count\_\_c.

5. MasterEventTrigger

**Description:**

Apex trigger for the Event object. This Trigger is using a helper class called ClassEventHandler.

**Business Functionality:**

On before insert and update events is calling the addClientStatus() method from the ClassEventHandler apex class that updates the Event.Client\_Status\_\_c with the related Lead's or Contact' s Account Client Status.

On  before update and after insert events is calling the LastContactActivityDate() method from the ClassEventHandler apex class that calculates the related Lead's or Contact's Last\_Activity\_Date\_\_c.

On after insert and update events is calling the SendNotificationEmail() method from the ClassEventHandler apex class that sends an email with the Event summary information to a selected list of contacts if the Send\_Notification\_Email\_\_c checkbox is set to true

**Affected Salesforce Fields:**

Client\_Status\_\_c, Lead.Last\_Activity\_Date\_\_c, Contact.Last\_Activity\_Date\_\_c, Send\_Notification\_Email\_\_c

6. MasterTaskTrigger

**Description:**

Apex trigger for the Task object. This Trigger is using a helper class called ClassTaskHandler.

**Business Functionality:**

On before insert and update events is calling the addClientStatus() method from the ClassTaskHandler apex class that updates the Task.Client\_Status\_\_c with the related Lead's or Contact' s Account Client Status.

On  before update and after insert events is calling the LastContactActivityDate() method from the ClassTaskHandler apex class that calculates the related Lead's or Contact's Last\_Activity\_Date\_\_c.

On after insert and update events is calling the SendNotificationEmail() method from the ClassTaskHandler apex class that sends an email with the Task summary information to a selected list of contacts if the Send\_Notification\_Email\_\_c checkbox is set to true

**Affected Salesforce Fields:**

Client\_Status\_\_c, Lead.Last\_Activity\_Date\_\_c, Contact.Last\_Activity\_Date\_\_c, Send\_Notification\_Email\_\_c

7. MasterApplicationTrigger

**Description:**

Apex trigger for the Application object. This Trigger is using a helper class called ClassApplicationHandler.

**Business Functionality:**

On after insert events is calling the linkContactidApplication() method from the ClassApplicationHandler apex class that is setting the Opportunity\_id\_\_c and Contact\_id\_\_c id fields based on the Opportunityid\_\_c and Contactid\_\_c string fields.

**Affected Salesforce Fields:**

Opportunity\_id\_\_c,  Contact\_id\_\_c

8. MasterAgreementTrigger

**Description:**

Apex trigger for the Agreement object. This Trigger is using a helper class called ClassAgreementHandler.

**Business Functionality:**

On after insert events is calling the linkContactidAgreement() method from the ClassAgreemnetHandler apex class that is setting the Applicationid\_\_c and Projectid\_\_c, Contactid\_\_c id fields based on the  Application\_id\_\_c, Project\_id\_\_cand Contact\_id\_\_c string fields.

**Affected Salesforce Fields:**

Applicationid\_\_c ,Projectid\_\_c, Contactid\_\_c

Apex Scheduled Classes:

* all the test classes are being named after the main apex classes with "\_Test" appended at the end

1. retrieveGAdata

**Description:**

Apex class that sends get requests to the Google Analytics server and gets an access token. Using the access token it creates another http request with some contact and lead related information that occurred withing the last day. The matched fields are being inserted in Salesforce under the related contacts and leads matched by the MBACOID\_\_C field. The class is being scheduled to run on a daily basis and the process name is entitled "GA push contact/lead data to SF".

**Affected Salesforce Fields:**

A\_Days\_Since\_Last\_Session\_\_c, GA\_Campaign\_\_c, GA\_City\_\_c , GA\_Country\_\_c, GA\_Social\_\_c, GA\_Source\_Medium\_\_c, GA\_Sessions\_\_c, GA\_Session\_Duration\_\_c

2. LeadActivityBatch

**Description:**

Apex batch class that sets the Lead status to Back To Marketing for the Prequalified and Qualified Leads that had no activity in the past 14 days. The class is scheduled to run on a daily basis and the scheduled job is being called : "Lead Activity Batch - 1pm run".

**Affected Salesforce Fields:**

Lead.Status

3. LeadActivityBatch2

**Description:**

Apex batch class that sets the Lead status from  Sales Development Lead to  Pre Qualified Leads if they have not been contacted for 15 days. The class is scheduled to run on a daily basis and the scheduled job is being called : "Lead Activity Batch2 - Back to PQL-".

**Affected Salesforce Fields:**

Lead.Status

4. OpportunityCloseDateReminderBatch

**Description:**

Apex  batch class to notify the opportunity owner if the related Project is due to close in the next day , sending an email with all the relevant information. The class is scheduled to run on a daily basis and the scheduled job is being called : "Opportunity Close Date Reminder - 12pm everyday".

**Affected Salesforce Fields:**

- no fields affected-

Apex Helper Classes

1. CreateTestObjects

**Description:**

Apex helper class that creates bulk SObjects in order to ease the building process of test classes.

**Available Methods:**

 CreateLeads (Integer numberofleads,String name, Account a)

* creates a set number of leads with a specified name related to an input Account object

 CreateAccounts(Integer numberofaccounts,String name)

* creates a set number of accounts with a specified account name

 CreateContacts(Integer numberofcontacts,String name, Account a)

* creates a set number of contacts with a specified name related to an input Account object

 CreateOpportunities(Integer numberofopportunities,String name, Account a,Contact c, String stagename)

* creates a set number of opportunities with a specified Name and Stage related to an input Account and Contact object

 CreateTasks(Integer numberofTasks,String name,Contact c, Lead l)

* creates a set number of tasks with a specified Name related to an input Contact and Lead object

 CreateEvents(Integer numberofEvents,String name, Contact c , Lead l)

* creates a set number of events with a specified Name related to an input Contact and Lead object

 CreateApplications(Integer numberofapplications,String name, List<Contact> contacts , List<Opportunity> opportunities)

* creates a set number of applications with a specified Name related to an input list of Contacts and Opportunities

CreateAgreements(Integer numberofagreements,String name, List<Contact> contacts ,List<application\_\_c> applications, List<Opportunity> opportunities)

* creates a set number of agreements with a specified Name related to an input list of Contacts, Applications and Opportunities

2. RecursionControl

**Description:**

Simple apex class that is being used to check for recursion issues in the trigger context environment

Apex Handler Classes

* all the triggers mentioned above have their functionality encapsulated in a apex class per trigger

1.ClassOpportunityHandler

**Description:**

Handler class for the MasterOpportunityTrigger

**Available methods:**

*addProductWhenStageIsPendingPayment()* **- method depreciated-**

* *creates automatically products when the opportunity stage reaches 'In Progress'*

addOpportunityToCampaign()

* adds the repeat opportunities to the "Inbound – Repeat" campaign

changeOppStatus()

* sets the automatic transition of the Project\_Stage\_\_c picklist values based on the current StageValue of the Opportunity

OppCountOnContact()

* calculates the closed, open and created opportunities for each Contact

setSalesManager()

* sets the opportunity owner to the matched name from the Delivery\_Owner\_\_c field

updateOpportunityCloseDate()

* updates the Opportunity CloseDate to 7 days later if the CloseDate is Today  and the Opportunity is not closed yet.

*RapidMinerFields()* **- method depreciated -**

* *updates related rapidminer fields*

CountActivitiesOnOpportunities()

* counts the related emails, meetings, calls related to an opportunity

2.ClassAccountHandler

**Description:**

Handler class for the MasterAccountTrigger

**Available methods:**

newAccountDuplicateCheck()

* checks if the inserted accounts by a non-admin user already exist in salesforce if so if the user then the missing information with regards to the Primary Sector, Company Size and HQ Locations are being filled in for the existing account and the new account is deleted. The user is being informed by the changes occurred through an email.

AccountQualityScore()

* computes a account information quality score based on the completeness of the related fields

3.ClassLeadHandler

**Description:**

Handler class for the MasterLeadTrigger

**Available methods:**

QualifiedLeadStatus()

* if the lead is i neither of the following stages:"Pre Qualified  Lead" , "Sales Developement Lead",  "Return To Marketing" and the Lead Score is >=20 then the Lead Stage becomes "Qualified Lead"

LeadGenQualifiedLeads()

* if a lead is being created by a "Lead Generation Profile" and the lead Status is "Pre Qualified Lead" then the leads become "Qualified Lead"

LeadQualityScore()

* computes a lead information quality score based on the completeness of the related fields

LinkLeadToAccount()

* upon the insertion of new leads that do not have an Account object linked against , it matches the Company\_\_c string field from the Lead object against the first found Account Name and sets the Account\_\_c to the related AccountId ( Note: if no Accounts are found then no changes are being performed)

*SuggestedEmail()* **- method depreciated -**

* *computes a suggested Lead email based on the Lead\_Email\_Format\_\_c and Lead\_Email\_Ending\_\_c values*

4.ClassContactHandler

**Description:**

Handler class for the MasterContactTrigger

**Available methods:**

GaCountryToContact()

* updated the Country\_\_c field with the updated GA\_Country\_\_c field

ContactQualityScore()

* computes a contact information quality score based on the completeness of the related fields

ContactsAtAccount()

* calculates the number of contacts that created opportunities, contacts that closed opportunities and total number of Contacts related to a account

*Contact\_Email\_Format()***- method depreciated -**

* *computes the email format model of the contact based on some already defined models*

*Account\_Email\_Format ()***- method depreciated-**

* *computes the general email format of the account based on the format frequency of all the contacts related to a account*

5.ClassEventHandler

**Description:**

Handler class for the MasterEventTrigger

**Available methods:**

addClientStatus()

* updates the Event.Client\_Status\_\_c field based on the related contacts or leads Client\_Status\_\_c

LeadActivityCount()

* updates the Activity\_Count\_\_c field with the number of related events for the Lead Object

LastContactActivityDate()

* updates the Last\_Activity\_Date\_\_c on the related Leads or Contact objects based on the last event ActivityDate field

*UpdateLeadStatusIfContacted()***- method depreciated -**

* *updates the lead status to 'Working - Contacted' if there are any events related to it and the prior lead status is either "Open - Not Contacted"  or  "Marketing suspect"*

sendNotificationEmail()

* sends a summary information email with the related event information to the event owner and to a selected email from a series of picklist values if the Send\_Notification\_Email\_\_c checkbox is ticked

6.ClassTaskHandler

**Description:**

Handler class for the MasterTaskTrigger

**Available methods:**

addClientStatus()

* updates the Task.Client\_Status\_\_c field based on the related contacts or leads Client\_Status\_\_c

LeadActivityCount()

* updates the Activity\_Count\_\_c field with the number of related tasks for the Lead Object

LastContactActivityDate()

* updates the Last\_Activity\_Date\_\_c on the related Leads or Contact objects based on the last task ActivityDate field

*UpdateLeadStatusIfContacted()***- method depreciated -**

* *updates the lead status to 'Working - Contacted' if there are any tasks related to it and the prior lead status is either "Open - Not Contacted"  or  "Marketing suspect"*

sendNotificationEmail()

* sends a summary information email with the related task information to the event owner and to a selected email from a series of picklist values if the Send\_Notification\_Email\_\_c checkbox is ticked

7.ClassApplicationHandler

**Description:**

Handler class for the MasterApplicationTrigger

**Available methods:**

linkContactidApplication()

* upon the insertion of new applications links the related Opportunity and Contact based on the Contactid\_\_c and Opportunityid\_\_C string fields

AppCountOnOpportunity()

* calculates the total number of applciations related to a Opportunity

8.ClassAgreementHandler

**Description:**

Handler class for the MasterAgreementTrigger

**Available methods:**

linkContactidAgreement()

* upon the insertion or update of the agreement objects it links them to the related Opportunities, Contacts and Applications based on the Project\_id\_\_c, Application\_id\_\_c and Contact\_id\_\_c string fields