Scenario :1

=====================================================================================

1. Object :Opportunity

a. When Existing Opportunity with statge name as 'Closed Won' is

updated to any other stage it has throw error message

b. When existing Opportunity record stageName is changed to

closed won then recalculate the Amount\_Won for the Correspondng Account (Which is a sum of all closed opportunities)

c. When ever the stage name is changed to Closed and won

share that opportunity record with user 'karthic'

public class ClosedOptHandler {

public static void beforeUpdate(Map<Id,Opportunity> oldMap,Map<Id,Opportunity> newMap){

Set<Id> accountIds=new Set<Id>();

for(Id a:oldMap.keySet()){

Opportunity newOpt=newMap.get(a);

Opportunity oldOpt=oldMap.get(a);

if(oldOpt.stageName=='Closed Won' && newOpt.stageName!='Closed Won'){

newOpt.addError('Stage Name cannot be updated');

}

}

}

public static void afterUpdate(Map<Id,Opportunity>oldMap,Map<Id,Opportunity> newMap){

Set<Id> accountIds=new Set<Id>();

for(Id a:oldMap.keySet()){

Opportunity newOpt=newMap.get(a);

Opportunity oldOpt=oldMap.get(a);

if(newOpt.StageName=='Closed Won'){

accountIds.add(newOpt.accountId);

}

}

if(accountIds.size()>0){

List<Account> accs=[select Amount\_Won\_\_c ,(select Amount from Opportunities where StageName='Closed Won') from Account where Id in:accountIds ];

Decimal sum=0;

for(Account a:accs){

for(Opportunity p:a.opportunities){

sum=sum+p.amount;

}

a.Amount\_Won\_\_c=sum;

}

update accs;

}

}

public static void oppShare(List<Opportunity> newOpt){

User u=[select id from User where alias='kart'];

List<OpportunityShare> share=new List<Opportunityshare>();

for(Opportunity op: newOpt){

if(op.stageName=='Closed Won'){

OpportunityShare ops=new OpportunityShare();

ops.opportunityId=op.id;

ops.OpportunityAccessLevel='Edit';

ops.UserOrGroupId=u.id;

ops.RowCause='Manual';

share.add(ops);

}

}

insert share;

}

}

------------------------------------------------------------------------------------------------------------------------------------------

Trigger :

trigger closedOppty on Opportunity (after insert,before update,after update) {

if(Trigger.isAfter && Trigger.isInsert){

ClosedOptHandler.oppShare(Trigger.new);

}

if(Trigger.isBefore && Trigger.isUpdate){

ClosedOptHandler.beforeUpdate(Trigger.oldMap, Trigger.newMap);

}

if(Trigger.isAfter && Trigger.isUpdate){

ClosedOptHandler.afterUpdate(Trigger.oldMap,Trigger.newMap);

ClosedOptHandler.oppShare(Trigger.new);

}

}

Scenario 2:

Opportunity :

Create a Custm Fields : Field Name DataType

---------------------------------------

Total\_Opportunities\_\_c : Number

Total\_Opportunity\_Amount\_\_c : Currency

a .When ever a new Opportunity is creatd recalculate the total opportunites and Total Opportunity

amount for the corresponding Account

b. When a new Opportunity is created with out entering the account value throw error

trigger opptScope on Opportunity (before insert, after insert) {

if(Trigger.isBefore && Trigger.isInsert){

for(Opportunity opp=Trigger.new){

if(opp.accountId==null){

opp.addError('Opportuity Can not created with out AccountId');

}

}

}

if(Trigger.isAfter && Trigger.isInsert){

Set<Id> accId=new Set<Id>();

for(Opportunity op:Trigger.New){

accId.add(op.accountId);

}

List<Account> accounts=[select Total\_Opportunities\_\_c,Total\_Opportunity\_Amount\_\_c,

(select id,Amount from Opportunities) from Account where Id in:accId];

for(Account a:accounts){

a.Total\_Opportunities\_\_c=a.opportunities.size();

Decimal sum=0;

for(Opportunity p:a.opportunities){

sum=sum+p.amount;

}

a.Total\_Opportunity\_Amount\_\_c=sum;

}

update accounts;

}

}

====================================================================================

Scenario 3:

=====================================================================================

Object : Contact

Custom Fields : Field Name DataType

----------------------------------------------

Source PickList: Sfdc

SAPExternal

Account :

Custom Fields : Field Name DataType

----------------------------------------------

Source PickList: Sfdc

SAPExternal

a. When we are trying to delete any contact record whoes source is SAP External it should throw error

b. When we trying to delete a contact whose account source is SAP External it has to throw error

trigger conDeletes on Contact (before delete) {

//List<Contact> cons=[select Source\_\_c ,Account.Source\_\_c from Contact where Id in:Trigger.Old];

for(Contact c:Trigger.old){

if(c.Source\_\_c=='SAP External' || c.Account.Source\_\_c=='SAP External'){

c.addError('You can not delete Conatct');

}

}

}

=====================================================================================

Scenario 4:

=====================================================================================

Object : Opportunity :

1. When ever a new Opportunity is created with Amount more than 10 lakhs and source as 'Web'

Then add karthic as OpportunityTeamMember

Solution :

trigger optyTeam on Opportunity (after insert) {

User u=[select id from User where alias='kart'];

List<OpportunityTeamMember> teams=new List<OpportunityTeamMember>();

for(Opportunity op:Trigger.New){

if(op.amount> 1000000 && op.LeadSource=='Web'){

OpportunityTeamMember ot=new OpportunityTeamMember();

ot.OpportunityId=op.id;

ot.OpportunityAccessLevel='Edit';

ot.userId=u.id;

ot.TeamMemberRole='Sales Manager';

teams.add(ot);

}

}

insert teams;

}

=====================================================================================

Scenario 5:

=====================================================================================

Object : Opportunity

1. When ever the opportunity stage is changed create dummy record in the PipeLine object with

following fieldMaping

PipleLine\_\_c p=new PipeLine\_C

p.Stage=Opportunity\_OldStage---Opportunity\_NewStage

p.Days =Today()-opportunity\_LastmOdifiedDate

p.Name=opportunityName

Solution :

trigger optyStageChange on Opportunity (after update) {

List<PipeLine\_\_c> optList=new List<PipeLine\_\_c>();

for(Id opId:Trigger.OldMap.keySet()){

Opportunity oldOpt=Trigger.OldMap.get(opId);

Opportunity newOpt=Trigger.newMap.get(opId);

if(oldOpt.stageName!=newOpt.StageName){

PipeLine\_\_c p=new PipeLine\_\_c();

p.Name=newOpt.Name;

p.Stage\_\_c=oldOpt.StageName+'-'+newOpt.StageName;

Date modDate=oldOpt.LastModifiedDate.date();

Integer days=System.Today().daysBetween(modDate);

p.Days\_\_C=days;

p.ConvertedDate\_\_c=System.today();

optlist.add(p);

}

}

insert optlist;

}

=====================================================================================

Scenario :6

=====================================================================================

Object : Application\_\_c

FieldName DataType Options

------------------------------------------------------

First Name Text -

Type PickList New Card

Cancel Card

Applicant PickList Business

Salaried

Status PicKList Pending

Approved

Rejected

Executive Lookup(user)

Email Email -

Object : BlackList

Field Name DataType

------------------------------------

Name Text

Object : Customer

Field Name DataType Options

-----------------------------------------------------

First Name Text

Applicatant PickList Business

Salaried

Workflow : When a new Applciation is creatd with

Type =new Card

Status=Pending

Action : Task : Assing to the Owner

Approval : When a ever a application is submitted for Approval

Entry Criteria : Type =New Card

Automated Approved : Submited Executives Manager

Approval steps : All Applications should enter step 1

Approval Action : Field Update Status= Approved

Rejected Action : Field Update Status ='Rejected'

1. When a new Application is submited for Approvel with Application type as 'New Card'

a. check wheather Applicant name in black list object

b. If name is in the black list object Then update the status to Rejected

2. When applicant name is not in the black list submit the record for approval using trigger

and set the status as pending

3. When ever the application Status is changed to Approved

a. Create a new Customer record based on application details

Customer FirstName =Application Firstname

Customer Type =Application -ApplicantType

4.When ever the application status is rejected

a. send a email to Executive ,Executive manager, Applicant and user with role : Verification