35. Context Variables and Delete Operation - 19 May 2022

Context Variables:

1] trigger.new 🡺 It returns a List of new version of the records (List<Object>).

2] trigger.newMap 🡺 It returns a Map of new version of the records (Map<Id,Object>).

3] trigger.old 🡺 It returns a List of old version of the records (List<Object>).

4] trigger.oldMap 🡺 It returns a Map of Old version of the records (Map<Id,Object>)

5] trigger.isInsert 🡺 Operation is of Insert

6] Trigger.isUpdate 🡪 Operation is of Update

7] trigger.isDelete 🡪 Operation is of Delete.

8] trigger.isUndelete 🡪 Operation is of Undelete

9] trigger.isBefore 🡪 Event is of Before operation.

10] trigger.isAfter 🡪 Event is of After operation.

Delete Operation:

User Story 1] Create an Object “Opportunity Backup”. Take Backup of Opportunity when a record gets deleted.

|  |  |
| --- | --- |
| Event | Before |
| Operations | Delete |
| Trigger Object | Opportunity |
| Affecting Object | Opportunity Backup |

Bakwas Code / Garib Code

trigger OpportunityTrigger on Opportunity (before delete) {

for(Opportunity objOpp : trigger.old){//Silver Microsystem

Opportunity\_Backup\_\_c objOppBK = new Opportunity\_Backup\_\_c(Opp\_Name\_\_c= objOpp.Name, Type\_\_c=objOpp.Type, Closed\_Date\_\_c=objOpp.CloseDate);

insert objOppBK;

}

}

DML Statement LIMIT: Insert, Update, Delete, Undelete

We can perform maximum 150 DML Statements per transaction.

System.LimitException: Too Many DML Statements: 151

Best Tarika:

trigger OpportunityTrigger on Opportunity (before delete) {

List<Opportunity\_Backup\_\_c> oppBkList = new List<Opportunity\_Backup\_\_c>();

for(Opportunity objOpp : trigger.old){//160

Opportunity\_Backup\_\_c objOppBK = new Opportunity\_Backup\_\_c(Opp\_Name\_\_c= objOpp.Name, Type\_\_c=objOpp.Type, Closed\_Date\_\_c=objOpp.CloseDate);

oppBkList.add(objOppBK);

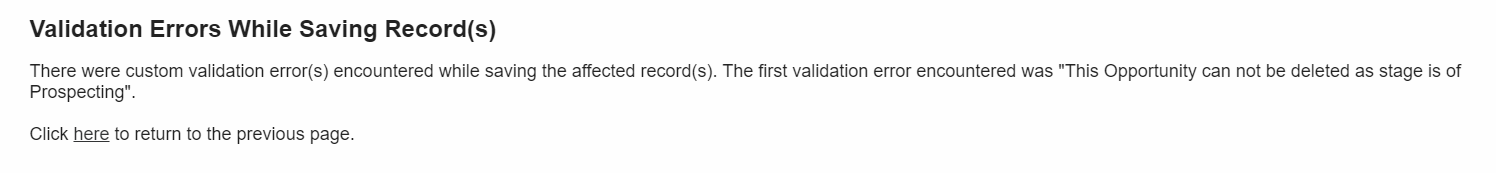
}

if(!oppBkList.isEmpty())

Database.insert(oppBkList,false);

}

User Story 2] Show Error, if user deletes an Opp record of Stage “Prospecting”. Else Take its Backup.



trigger OpportunityTrigger on Opportunity (before delete) {

List<Opportunity\_Backup\_\_c> oppBkList = new List<Opportunity\_Backup\_\_c>();

for(Opportunity objOpp : trigger.old){

if(objOpp.StageName =='Prospecting'){

objOpp.addError('This Opportunity can not be deleted as stage is of Prospecting');

}

else{

Opportunity\_Backup\_\_c objOppBK = new Opportunity\_Backup\_\_c(Opp\_Name\_\_c= objOpp.Name, Type\_\_c=objOpp.Type, Closed\_Date\_\_c=objOpp.CloseDate);

oppBkList.add(objOppBK);

}

}

if(!oppBkList.isEmpty())

Database.insert(oppBkList,false);

}

User Story 3] Prevent Opp Deletion if,

- Amount is greater or equals to 18000

- Closed Date Month is August

- Stage is “Prospecting” or “Closed Lost”.

Else, delete the record and take backup.

trigger OpportunityTrigger on Opportunity (before delete) {

List<Opportunity\_Backup\_\_c> oppBkList = new List<Opportunity\_Backup\_\_c>();

for(Opportunity objOpp : trigger.old){

if(objOpp.Amount >= 18000 && objOpp.CloseDate.month() == 8 && (objOpp.StageName=='Prospecting' || objOpp.StageName=='Closed Lost') ){

objOpp.Amount.addError('This Opp can not be deleted...!!!');

}

else{

Opportunity\_Backup\_\_c objOppBK = new Opportunity\_Backup\_\_c(Opp\_Name\_\_c= objOpp.Name, Type\_\_c=objOpp.Type, Closed\_Date\_\_c=objOpp.CloseDate);

oppBkList.add(objOppBK);

}

}

if(!oppBkList.isEmpty())

Database.insert(oppBkList,false);

}

Email Code

Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();

mail.setToAddresses(new String[] {'silvermicrosystems@gmail.com'});

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Welcome to Air India');

mail.setPlainTextBody('Hi, Welcome to Air India ');

Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });

User Story 4] Send an Email to [silvermicrosystems@gmail.com](mailto:silvermicrosystems@gmail.com) when a opp record gets deleted.

|  |  |
| --- | --- |
| Event | after |
| Operations | Delete |
| Trigger Object | Opportunity |
| Affecting Object | --NA-- |

trigger OpportunityTrigger on Opportunity (after delete) {

List<Messaging.SingleEmailMessage> mailList = new List<Messaging.SingleEmailMessage>();

for(Opportunity objOpp : trigger.old){

Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();

mail.setToAddresses(new String[] {'silvermicrosystems@gmail.com'});

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Record Deleted');

mail.setPlainTextBody('Hello, \n Opp '+objOpp.Name+' record has been deleted');

mailList.add(mail);

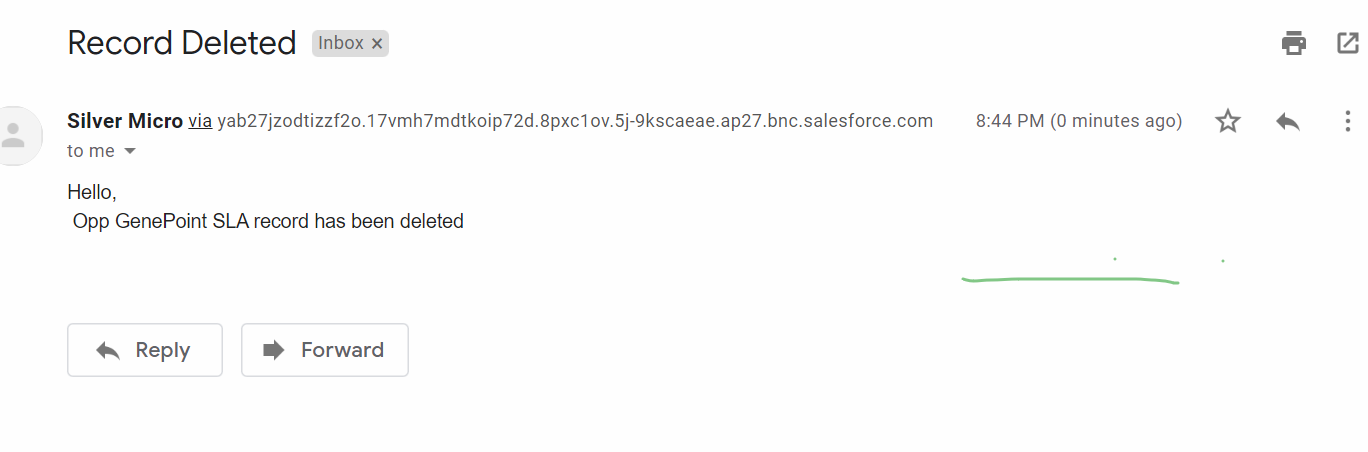
}

if(!mailList.isEmpty()){

Messaging.sendEmail(mailList,false);

}

}



Assignments:

1] Take backup of APPLICANT object. Only if Applicant was eligible for Police Verification, DOB month is July or Nov and Gender is “Male” or “Female”.

Else prevent it.

2] Create / Update a Case record. If Priority is “High”, then Case Violation should be “Yes”.