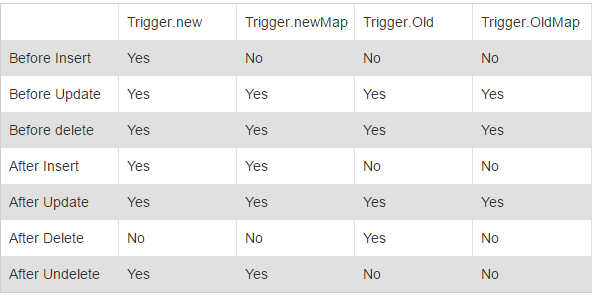
36. Update Operations and OldMap - 20 May 2022



1] Send an Email when Account Rating is updated to “Hot”.

|  |  |
| --- | --- |
| Event | After |
| Operations | Update |
| Trigger Object | Account |
| Affecting Object | --NA-- |





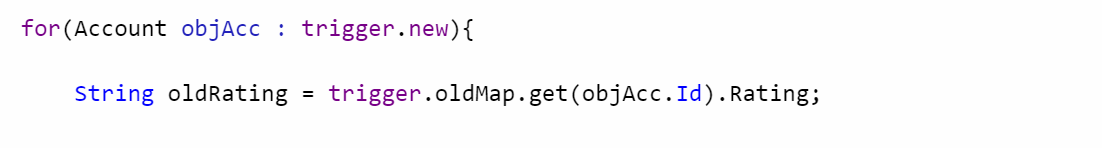
Concept of Map

Map<Integer,String> studentMap = new Map<Integer,String>();

studentMap.put(100,’Bunty’);

studentMap.put(200,’Bubli’);

studentMap.get(100); //Bunty



Account Record : 0015j00000f5tbQ , Big B1

|  |  |
| --- | --- |
| Trigger.new | Trigger.oldMap |
| Hot | Cold |

Dusra/Another Example:

Account : Cinemax : 123 🡪 Warm , Hot

Poonam Mall : 402 🡪 Cold , Warm

Eternity Mall : 604 🡪 Warm Cold

|  |  |  |
| --- | --- | --- |
| Account | Trigger.oldMap | Trigger.new |
| 123 : Cinemax | Warm | Hot |
| 402 : Poonam Mall | Cold | Warm |
| 604 : Eternity Mall | Warm | Cold |

Trigger.old 🡪 List<Account>

|  |
| --- |
| Cinemax |
| Poonam Mall |
| Eternity Mall |

Trigger.oldMap 🡪 Map<Id, Account>

|  |  |
| --- | --- |
| ID : 123 | Cinemax |
| ID : 402 | Poonam Mall |
| ID: 604 | Eternity Mall |

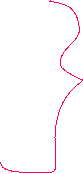
for(Account objAcc : trigger.new){

for(Account objAccOld : trigger.old ){

if(objAcc.Name == objAccOld.Name){



if(objAcc.Rating != objAccOld.Rating)



}

}

}

for(integer i=1; i<=10; i++){

for(integer j=1; j<=10; j++){



System.debug(i);



}

}

trigger AccountTrigger20May on Account (after update) {

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

for(Account objAcc : trigger.new){ // BigB List<Account> = Hot

//String newRating = objAcc.Rating; //Hot

// String oldRating = trigger.oldMap.get(objAcc.Id).Rating; //Cold

if(objAcc.Rating != trigger.oldMap.get(objAcc.Id).Rating){

if(String.isNotBlank(objAcc.Rating) && objAcc.Rating == 'Hot'){

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

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

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Account Rating Updated');

mail.setPlainTextBody('Account Rating Updated');

mailList.add(mail);

}

}

}

if(!mailList.isEmpty()){

Messaging.sendEmail(mailList);

}

}

Scenario 2: Send an Email when Account Rating is updated to “Hot” from “Cold”

New Value = Hot

Old Value = Cold

1st Tarika:

trigger AccountTrigger20May on Account (after update) {

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

for(Account objAcc : trigger.new){

String newRating = objAcc.Rating; //New Value

String oldRating = trigger.oldMap.get(objAcc.Id).Rating; //Old Value

if(newRating != oldRating){

if(newRating =='Hot' && oldRating=='Cold'){

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

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

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Account Rating Updated');

mail.setPlainTextBody('Account Rating Updated');

mailList.add(mail);

}

}

}

if(!mailList.isEmpty()){

Messaging.sendEmail(mailList);

}

}

Makkhan Code / Khatarnak Tarika:

trigger AccountTrigger20May on Account (after update) {

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

for(Account objAcc : trigger.new){

if(String.isNotBlank(objAcc.Rating)){

if(objAcc.Rating != trigger.oldMap.get(objAcc.Id).Rating){

if(objAcc.Rating =='Hot' && trigger.oldMap.get(objAcc.Id).Rating=='Cold'){

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

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

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Account Rating Updated');

mail.setPlainTextBody('Account Rating Updated');

mailList.add(mail);

}

}

}

}

if(!mailList.isEmpty()){

Messaging.sendEmail(mailList);

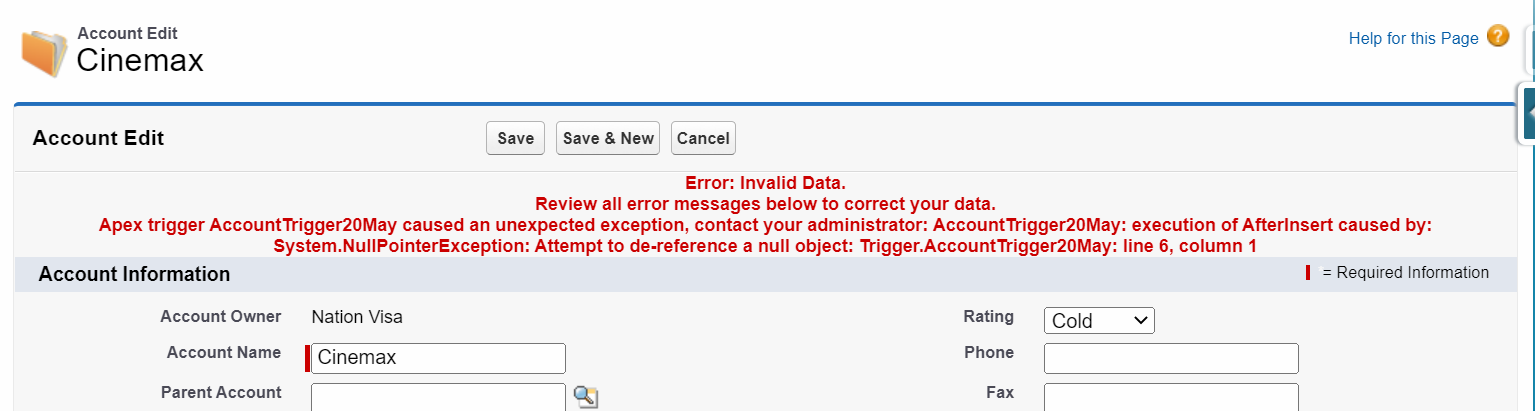
}

}

3] Send an Email

* When a new Account is Created.
* When an Account is updated to “Cold”.

|  |  |
| --- | --- |
| Event | After |
| Operations | Insert , Update |
| Trigger Object | Account |
| Affecting Object | --NA-- |



trigger AccountTrigger20May on Account (after insert, after update) {

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

for(Account objAcc: trigger.new){

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

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

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

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Welcome Message');

mail.setPlainTextBody('Account Record has been created..!!');

mailList.add(mail);

}

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

if(objAcc.Rating != trigger.oldMap.get(objAcc.Id).Rating){

if(objAcc.Rating == 'Cold'){

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

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

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Account Rating Updated');

mail.setPlainTextBody('Account Rating Updated');

mailList.add(mail);

}

}

}

}

if(!mailList.isEmpty()){

Messaging.sendEmail(mailList);

}

}

Assignment:

3] Send an Email

* When a new Account is Created.
* When an Account is updated to “Cold” from “Hot”.
* When an Account is deleted.

4] Create a new Applicant Record. Applicant should be eligible for Police Verification, if Gender is “Male”.

5] Send an Email When Pan Card is Updated.

