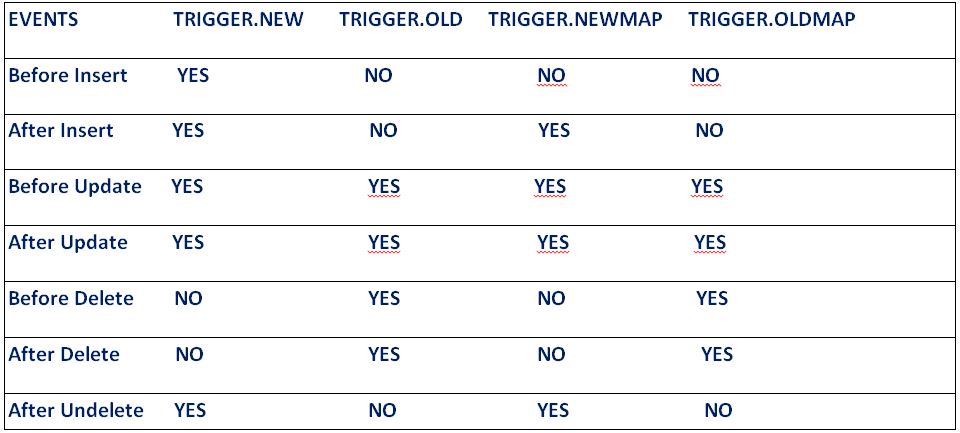
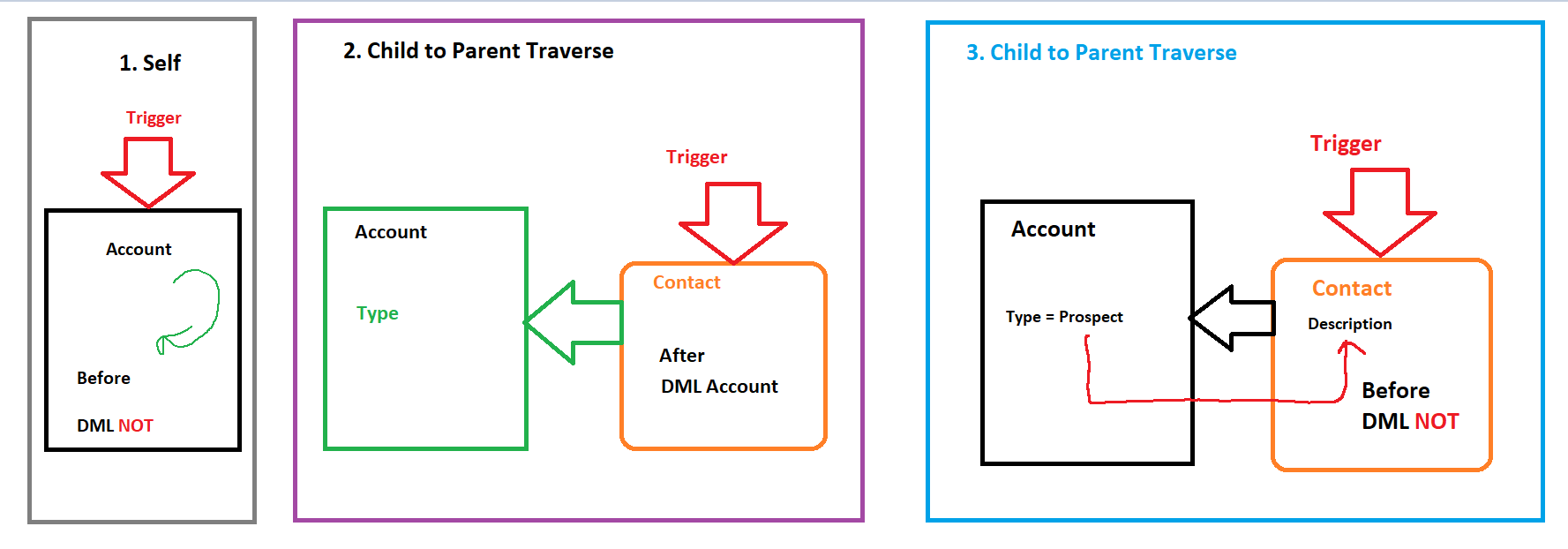
39. Child to Parent Traverse SELF - 26 May 2022







SAME STPES OF 🡪 Child to Parent Traverse

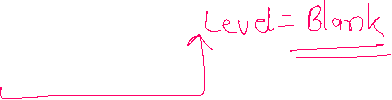
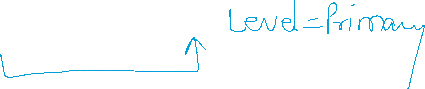
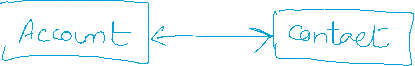
STEPS:

1] Create Set of Parent ID

2] Create Map<ID,ParentObject> 🡪 SQOL over parent for above SET and put the data in MAP.

3] Take Trigger Iteration again and Search key in MAP and apply the appropriate further logic.

User Story 1] Create / Edit the Contact. If It’s respective Account Rating is “Hot”, then Contact’s Level set to the “Primary”. Else keep blank.



|  |  |
| --- | --- |
| Event | Before |
| Operations | Insert, Update |
| Trigger Object | Contact |
| Affecting Object | Contact (**DML NO**) |

trigger ContactTrigger26May on Contact (before insert, before update) {

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

if( (trigger.isInsert || trigger.isUpdate) && trigger.isBefore){

for(Contact objCon : trigger.new){

if(objCon.AccountId !=null){

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

accIdSet.add(objCon.AccountId);

}

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

accIdSet.add(objCon.AccountId);

}

}

}

}

Map<Id,Account> accMap = new Map<Id,Account>();

if(!accIdSet.isEmpty()){

for(Account objAcc : [select Id,Rating from Account where Id IN : accIdSet ]){

accMap.put(objAcc.Id, objAcc);

}

}

if(!accMap.isEmpty()){

for(Contact objCon : trigger.new){

if(accMap.containsKey(objCon.AccountId)){

if(accMap.get(objCon.AccountId).Rating == 'Hot'){

objCon.Level\_\_c ='Primary';

}

else{

objCon.Level\_\_c ='';

}

}

}

}

}

User Story 2] Create / Edit the Contact. If It’s respective Account Rating is “Hot”, then Contact’s Level set to the “Primary”. If Rating is “Cold”, then Level set to the “Secondary”, If Rating is “Warm”, then Level = “Tertiary”. Else blank.

trigger ContactTrigger26May on Contact (before insert, before update) {

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

if( (trigger.isInsert || trigger.isUpdate) && trigger.isBefore){

for(Contact objCon : trigger.new){

if(objCon.AccountId !=null){

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

accIdSet.add(objCon.AccountId);

}

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

accIdSet.add(objCon.AccountId);

}

}

}

}

Map<Id,Account> accMap = new Map<Id,Account>();

if(!accIdSet.isEmpty()){

for(Account objAcc : [select Id,Rating from Account where Id IN : accIdSet ]){

accMap.put(objAcc.Id, objAcc);

}

}

if(!accMap.isEmpty()){

for(Contact objCon : trigger.new){

if(accMap.containsKey(objCon.AccountId)){

if(accMap.get(objCon.AccountId).Rating == 'Hot'){

objCon.Level\_\_c ='Primary';

}

else{

if(accMap.get(objCon.AccountId).Rating == 'Cold'){

objCon.Level\_\_c ='Secondary';

}

else{

if(accMap.get(objCon.AccountId).Rating == 'Warm'){

objCon.Level\_\_c ='Tertiary';

}

else{

objCon.Level\_\_c ='';

}

}

}

}

}

}

}

User Story 3] Create / Update An Address Record. If new city is “Nagpur” and old City was “Pune” , And its respective Applicant is “Eligible for Police Verification”, then copy Applicant’s PAN Card in the Address’s Pan Card field.

|  |  |
| --- | --- |
| Event | Before |
| Operations | Insert, Update |
| Trigger Object | Address |
| Affecting Object | Address (**DML NO**) |

trigger AddressTrigger26May on Address\_\_c (before insert, before update) {

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

if( (trigger.isInsert || trigger.isUpdate) && trigger.isBefore){

for(Address\_\_c objAdr : trigger.new){

if(objAdr.Applicant\_\_c!=null){

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

applicantIdSet.add(objAdr.Applicant\_\_c);

}

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

if(objAdr.City\_\_c != trigger.oldMap.get(objAdr.Id).City\_\_c){

applicantIdSet.add(objAdr.Applicant\_\_c);

}

}

}

}

}

Map<Id,Applicant\_\_c> applicantMap = new Map<Id,Applicant\_\_c>();

if(!applicantIdSet.isEmpty()){

for(Applicant\_\_c objApp : [select id,Police\_Verification\_\_c,Pan\_Card\_\_c from Applicant\_\_c where Id IN : applicantIdSet]){

applicantMap.put(objApp.Id, objApp);

}

}

if( (trigger.isInsert || trigger.isUpdate) && trigger.isBefore){

if(!applicantMap.isEmpty()){

for(Address\_\_c objAdr : trigger.new){

if(applicantMap.containsKey(objAdr.Applicant\_\_c)){

if(applicantMap.get(objAdr.Applicant\_\_c).Police\_Verification\_\_c){

if(trigger.isInsert && trigger.isBefore && objAdr.City\_\_c == 'Nagpur'){

objAdr.Applicant\_PAN\_CARD\_\_c = applicantMap.get(objAdr.Applicant\_\_c).Pan\_Card\_\_c;

}

else{

objAdr.Applicant\_PAN\_CARD\_\_c = '';

}

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

if(objAdr.City\_\_c == 'Nagpur' && trigger.oldMap.get(objAdr.Id).City\_\_c=='Pune'){

objAdr.Applicant\_PAN\_CARD\_\_c = applicantMap.get(objAdr.Applicant\_\_c).Pan\_Card\_\_c;

}

else{

objAdr.Applicant\_PAN\_CARD\_\_c = '';

}

}

}

else{

objAdr.Applicant\_PAN\_CARD\_\_c = '';

}

}

}

}

}

}

Assignment:

1] Create/Edit an Opp.

If Account Rating is “Hot” or “Warm”, then Opp Type should be “Existing Customer - Upgrade”, else None.

Copy Account’s SLA into Opp’s Description.

2] Create an Object Customer. Create a Field “Total Balance”.

Create an Object Transaction. Create Picklist Field “Transaction Type” (Deposit, Withdraw). And Currency Field “Amount”.

Where Customer and Transaction Object has Lookup relationship. (Customer 🡨 🡪 Transaction).

If Transaction Type is “Deposit” 🡪 Total Balance = Total Balance + Amount

If Withdraw 🡪 Total Balance = Total Balance – Amount.