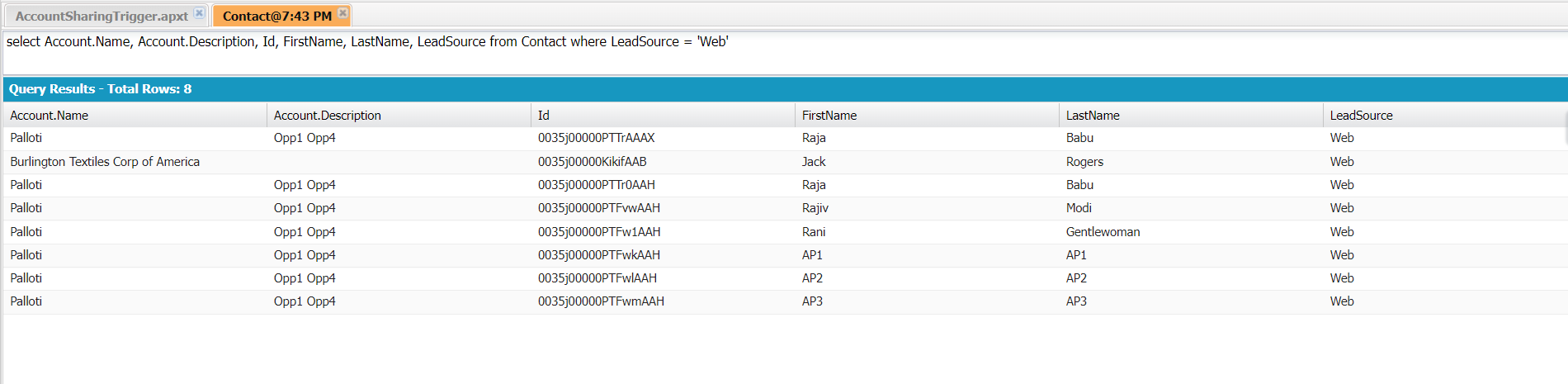
45. Child to Parent Traverse and Parent to Child Traverse - 07 June 2022

**User Story 1] If the Contact Lead Source is “Web”, then show “Contact FirstName, LastName and LeadSource as “Web” on Description field of Account.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account** | **Description** | **Contact First Name** | **Last Name** | **Lead Source** |
| **Cinemax** | **Bunty Modi Web** | **Bunty** | **Modi** | **Web** |
| **Bubli** | **Modi** | **Phone** |
| **Poonam Mall** | **Chintu Modi Web**  **Mintu Modi Web** | **Pintu** | **Modi** | **Phone** |
| **Chintu** | **Modi** | **Web** |
| **Mintu** | **Modi** | **Web** |

**select Account.Name, Account.Description, Id, FirstName, LastName, LeadSource from Contact where LeadSource = 'Web'**



global class BatchContactToAccount implements Database.Batchable<sObject>{

global Database.QueryLocator start(Database.BatchableContext BC){

return Database.getQueryLocator('select Account.Name, Account.Description,AccountId, Id, FirstName, LastName, LeadSource from Contact where LeadSource = \'Web\' ');

}

global void execute(Database.BatchableContext BC, List<Contact> conList){//trigger.new

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

if(!conList.isEmpty()){

for(Contact objCon : conList){

if(objCon.AccountId !=null)

accIDSet.add(objCon.AccountId);

}

}

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

if(!accIDSet.isEmpty()){

for(Account objAcc : [select Id,Description from Account where Id IN : accIDSet]){

accMap.put(objAcc.Id, objAcc);

}

}

if(!accMap.isEmpty()){

for(Contact objCon : conList){

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

if(String.isBlank(accMap.get(objCon.AccountId).Description)){

accMap.get(objCon.AccountId).Description = objCon.FirstName+' '+objCon.LastName+' '+objCon.LeadSource;

}

else{

accMap.get(objCon.AccountId).Description = accMap.get(objCon.AccountId).Description+'\n'+objCon.FirstName+' '+objCon.LastName+' '+objCon.LeadSource;

}

}

}

Database.update(accMap.values(), false);

}

}

global void finish(Database.BatchableContext BC){

}

}

TO RUN:

Database.executeBatch(new BatchContactToAccount());

Scenario 2] Show Address Name (ADR-0001), City, Pin Code in “List of Addresses” field on Applicant, such that,

- Address City should be “Nagpur”

- Pin Code starts with “44”

- Applicant should be eligible for Police Verification

global class BatchAddressToApplicant implements Database.Batchable<sObject> {

global Database.QueryLocator start(Database.BatchableContext BC){

return Database.getQueryLocator('select Name,City\_\_c, Pin\_Code\_\_c,Applicant\_\_r.List\_of\_Addresses\_\_c,Applicant\_\_c from Address\_\_c where City\_\_c=\'Nagpur\' and Pin\_Code\_\_c like \'44%\' and Applicant\_\_r.Police\_Verification\_\_c=true ');

}

global void execute(Database.BatchableContext BC, List<Address\_\_c> adrList){

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

if(!adrList.isEmpty()){

for(Address\_\_c objAdr : adrList){

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

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

}

}

}

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

if(!applicantIDSet.isEmpty()){

for(Applicant\_\_c objApp : [select Id,List\_of\_Addresses\_\_c from Applicant\_\_c where Id IN : applicantIDSet]){

applicantMap.put(objApp.id, objApp);

}

}

if(!applicantMap.isEmpty()){

for(Address\_\_c objAdr : adrList){

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

if(String.isBlank(applicantMap.get(objAdr.Applicant\_\_c).List\_of\_Addresses\_\_c)){

applicantMap.get(objAdr.Applicant\_\_c).List\_of\_Addresses\_\_c = objAdr.Name+' '+objAdr.City\_\_c+' '+objAdr.Pin\_Code\_\_c;

}

else{

applicantMap.get(objAdr.Applicant\_\_c).List\_of\_Addresses\_\_c = applicantMap.get(objAdr.Applicant\_\_c).List\_of\_Addresses\_\_c+'\n'+ objAdr.Name+' '+objAdr.City\_\_c+' '+objAdr.Pin\_Code\_\_c;

}

}

}

Database.update(applicantMap.values(),false);

}

}

global void finish(Database.BatchableContext BC){

}

}

TO RUN :

Database.executeBatch(new BatchAddressToApplicant());

PARENT TO CHILD TRAVERSE

1] Update All the Opportunities

* Delivery / Installation status as “Completed”, IF Account Rating “Hot”.
* Delivery / Installation status as “In progress”, IF Account Rating “Cold” OR “Warm”.
* Delivery / Installation status as “Yet to Begin”, otherwise.

Account 🡺 Opportunity

global class BatchAccountToOpporunity implements Database.Batchable<sObject>{

global Database.QueryLocator start(Database.BatchableContext BC){

return Database.getQueryLocator('select Id, Rating from Account');

}

global void execute(Database.BatchableContext BC, List<Account> accList){

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

if(!accList.isEmpty()){

for(Account objAcc : accList){

accMap.put(objAcc.Id, objAcc);

}

}

List<Opportunity> oppList = new List<Opportunity>();

if(!accMap.isEmpty()){

for(Opportunity objOpp : [select Id,AccountId,DeliveryInstallationStatus\_\_c from Opportunity where AccountId IN : accMap.keySet()]){

oppList.add(objOpp);

}

}

if(!oppList.isEmpty()){

for(Opportunity objOpp : oppList){

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

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

objOpp.DeliveryInstallationStatus\_\_c = 'Completed';

}

else{

if(accMap.get(objOpp.AccountId).Rating == 'Cold' || accMap.get(objOpp.AccountId).Rating=='Warm'){

objOpp.DeliveryInstallationStatus\_\_c = 'In progress';

}

else{

objOpp.DeliveryInstallationStatus\_\_c = 'Yet to begin';

}

}

}

}

Database.update(oppList,false);

}

}

global void finish(Database.BatchableContext BC){

}

}

TO RUN:

Database.executeBatch(new BatchAccountToOpporunity());

**Assignment:**

**1] Update All the**

* Address City as “Pune”, if Applicant Pan Card Starts with “P”.
* Address City as “Nagpur”, if Applicant Pan Card Starts with “N”.
* Address City as “Mumbai”, if Applicant Pan Card Starts with “M” AND Applicant is Eligible for Police Verification.

**2] Update All Applicant’s Gender as “Male”, for those records created in Between Date : 01 April 2022 to 07 June 2022.**

**NOTE -** List<Applicant\_\_c> appList = [select Id, First\_Name\_\_c, CreatedDate from Applicant\_\_c where CreatedDate>=2022-04-01T00:00:00Z and CreatedDate<=2022-06-07T00:00:00Z]