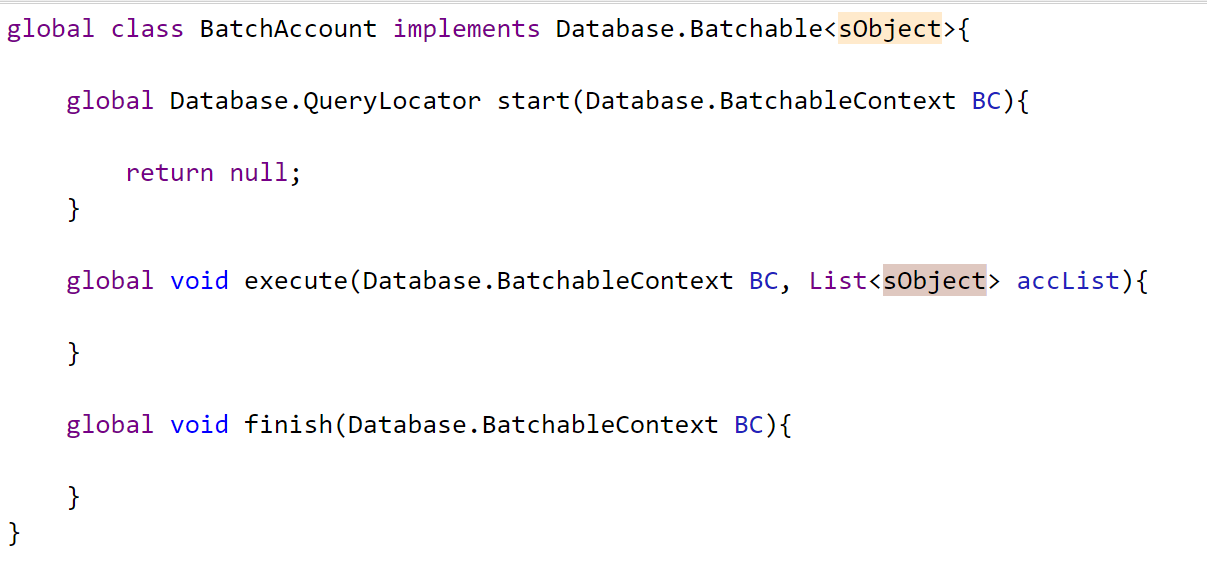
44. Basics of Batch Classes - 06 June 2022

**Definition**: When a normal class implements Database.Batchable<sObject> interface, then that class is known as “Batch Class”.

Syntax:







Method Purpose:

1] Start 🡺 SOQL Query

2] Execute 🡺 Business Logic

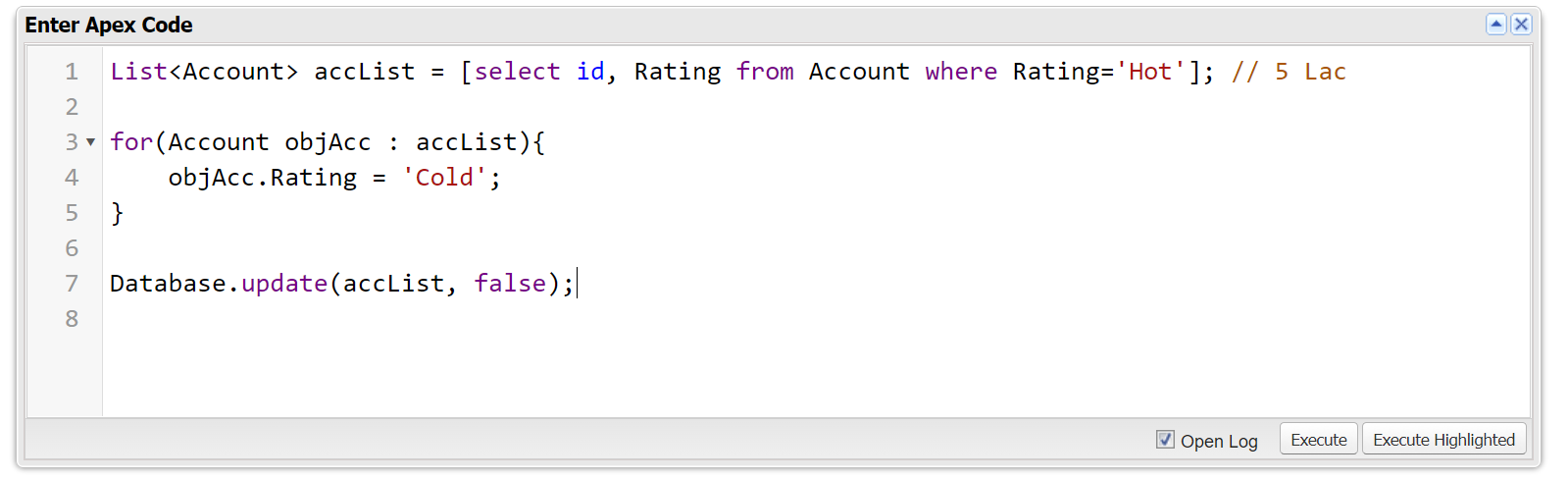
3] Finish 🡺 A] To Send Acknowledgment Mail

B] Batch Chaining

Why we use Batch Class?

It is used to process bulk Records.







User Story 1] Update All the Account Records from Rating “Hot” to “Cold”.

global class BatchAccount implements Database.Batchable<sObject>{

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

String query = 'select Id, Rating from Account where Rating=\'Hot\' ';

return Database.getQueryLocator(query); //07 Records

}

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

if(!accList.isEmpty()){

for(Account objAcc : accList){

objAcc.Rating = 'Cold';

}

Database.update(accList,false);

}

}

global void finish(Database.BatchableContext BC){

}

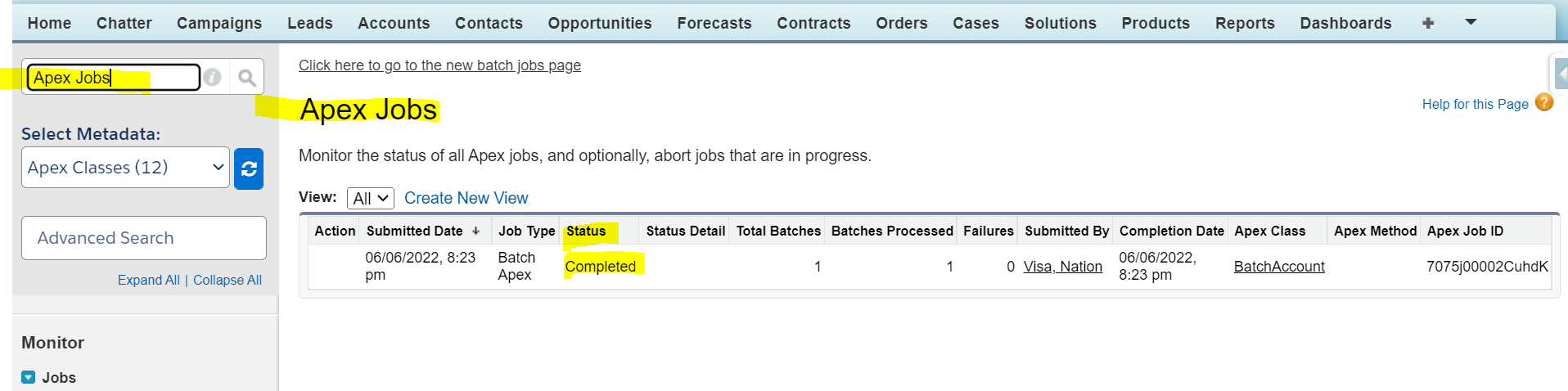
}

**TO RUN:**

BatchAccount objBatchAcc = new BatchAccount();

Database.executeBatch(objBatchAcc);

To Check The Status:

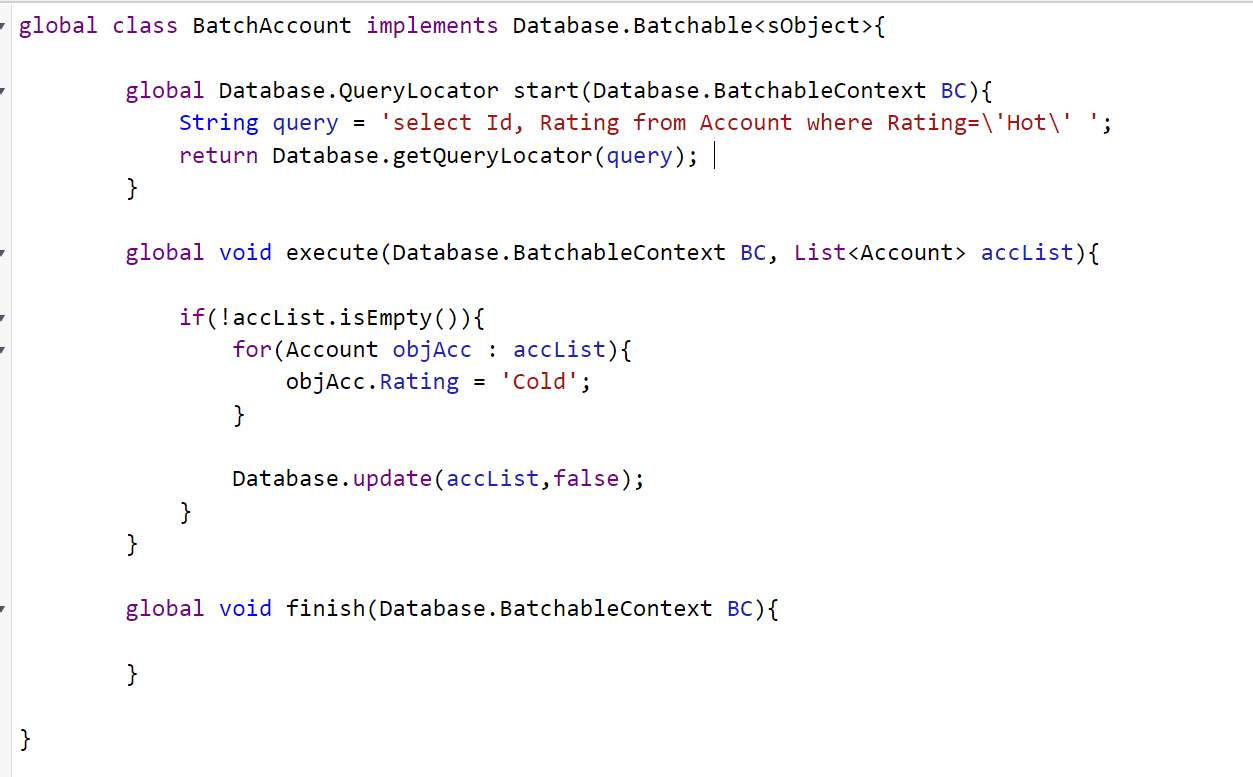


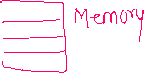
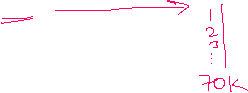
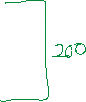
How Does It Work:

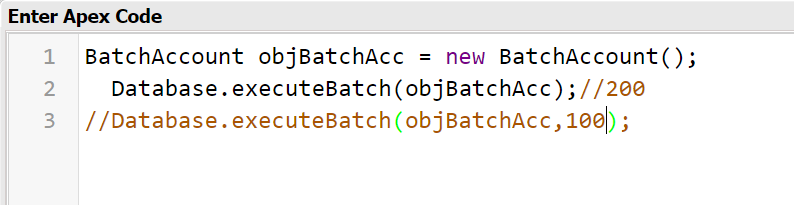
Batch Size Range:

Default Batch Size = **200**

Range = 1 - 2000









Execute Method Runs (N Times) = No. of Records / Batch Size

Example : Total Records 1 Lac, Default Batch Size = 200

Execute Run : 1 Lac / 200 = 500

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

Batch Size = 100

Example : Total Records 1 Lac, Batch Size = 100

Execute Run : 1 Lac / 100 = 1000

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

Batch Size = 2

Example : Total Records 1 Lac, Batch Size = 2

Execute Run : 1 Lac / 2 = 50000

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

Batch Size = 2000

Example : Total Records 1 Lac, Batch Size = 2000

Execute Run : 1 Lac / 2000 = 50

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

**Batch Size = 2500**

**Example : Total Records 1 Lac, Batch Size = 2500**

**Execute Run : 1 Lac / 2500 = NOT POSSIBLE**

User Story 2] If Applicant Gender is “Male” or “Female” and **NOT** eligible for Police Verification, Then Make it eligible for “Police Verification”.

global class BatchApplicant implements Database.Batchable<sObject>{

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

return Database.getQueryLocator('select Id, Gender\_\_c, Police\_Verification\_\_c from Applicant\_\_c where (Gender\_\_c=\'Male\' or Gender\_\_c=\'Female\' ) and Police\_Verification\_\_c = false ');

}

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

if(!applicantList.isEmpty()){

for(Applicant\_\_c objApp : applicantList){

objApp.Police\_Verification\_\_c = true;

}

Database.update(applicantList,false);

}

}

global void finish(Database.BatchableContext BC){

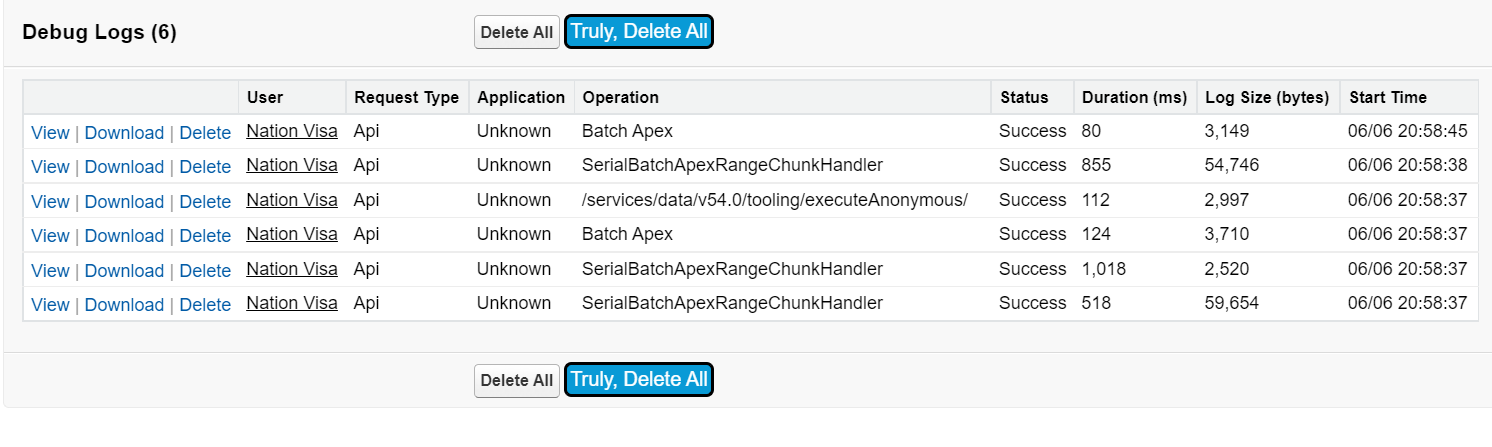
}

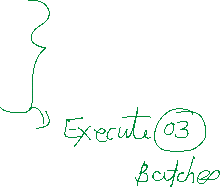
}

TO RUN :

BatchApplicant objBatchApp = new BatchApplicant();

Database.executeBatch(objBatchApp,7);





Sending Mail from Finish Method:

global class BatchApplicant implements Database.Batchable<sObject>{

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

System.debug('From Start Method');

return Database.getQueryLocator('select Id, Gender\_\_c, Police\_Verification\_\_c from Applicant\_\_c where (Gender\_\_c=\'Male\' or Gender\_\_c=\'Female\' ) and Police\_Verification\_\_c = true ');

}

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

if(!applicantList.isEmpty()){

for(Applicant\_\_c objApp : applicantList){

objApp.Police\_Verification\_\_c = false;

}

Database.update(applicantList,false);

System.debug('From Execute Method');

}

}

global void finish(Database.BatchableContext BC){

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

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:**

1] If Opportunity

Type = “Existing Customer - Update” and Closed Date Month is “April”, “June”, then Delivery Installation Status = “Completed”.

Type = “Existing Customer - Update” and Closed Date Month is “Jan”, “Dec”, then Delivery Installation Status = “Yet to Complete”.

Type = “New Customer”, then Delivery Installation Status = “In Progress”.

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