46. Batch Chaining - 09 June 2022

**Definition:** When a batch class is called from the finish method of another batch class, then the concept is known as “Batch Chaining”.



BatchClass2 {

start(){

}

execute(){

}

finish(){

}

}

BatchClass1 {

start(){

}

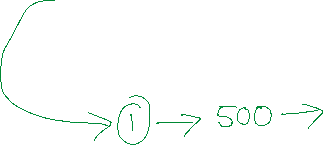
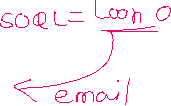
execute(){

}

finish(){

}

}



**User Story 1]** If Applicant Age < 18 or Age is null, then update Passport Message as “Your Passport has been rejected”.



After that, send mail to all the rejected Applicant’s that, your passport has been rejected.

**Batch Class 1:**

global class BatchApplicantPassportMessage implements Database.Batchable<sObject>{

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

return Database.getQueryLocator('select Id, Age\_\_c,Passport\_Message\_\_c from Applicant\_\_c where Age\_\_c < 18 or Age\_\_c=null ');

}

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

if(!appList.isEmpty()){

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

objApp.Passport\_Message\_\_c ='Your Passport has been rejected';

}

Database.update(appList, false);

}

}

global void finish(Database.BatchableContext BC){

Database.executeBatch(new BatchApplicantEmail(),10);

}

}

**Batch Class 2**

global class BatchApplicantEmail implements Database.Batchable<sObject> {

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

return Database.getQueryLocator('select id,First\_Name\_\_c, Passport\_Message\_\_c, Email\_Id\_\_c from Applicant\_\_c where Passport\_Message\_\_c=\'Your Passport has been rejected\' ');

}

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

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

if(!applicantList.isEmpty()){

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

if(!String.isBlank(objApp.Email\_Id\_\_c)){

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

mail.setToAddresses(new String[] {objApp.Email\_Id\_\_c});

mail.setSenderDisplayName('Silver Micro');

mail.setSubject('Urgent Message from Passport Office');

mail.setPlainTextBody('Hi '+objApp.First\_Name\_\_c+'\n'+objApp.Passport\_Message\_\_c);

mailList.add(mail);

}

}

if(!mailList.isEmpty()){

Messaging.sendEmail(mailList);

}

}

}

global void finish(Database.BatchableContext BC){

}

}

How many batch classes can be active at a time in a queue?

Ans: **05**

BatchClass 1 🡺 Active

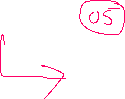
BatchClass 2 🡺

BatchClass 3 🡺 Active

BatchClass 4 🡺 Active

BatchClass 5 🡺

BatchClass 6 🡺 Active

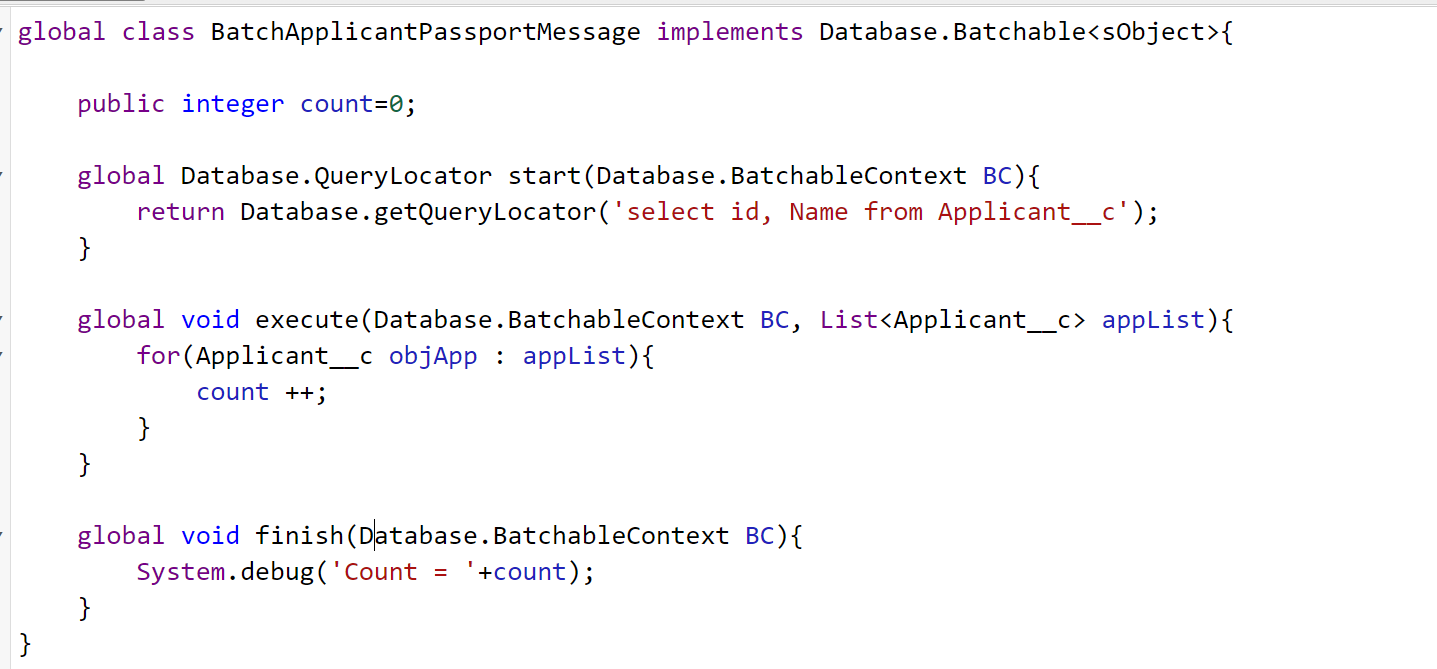


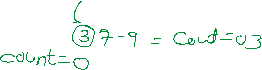
**Database.Stateful Interface**

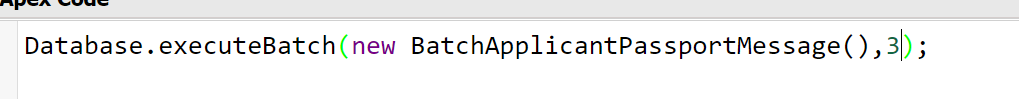
By Default, Batch class is stateless. That means, it can not maintain the state of its variable.

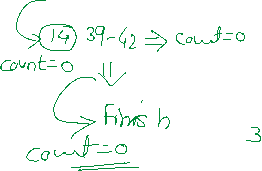
To make a batch class stateful, we use **Database.stateful** interface.

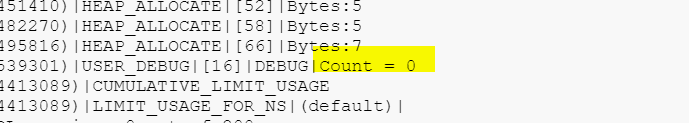
User Story 1] Fetch all the records of Applicant.











Assignment:

1] Update Police Verification to true for all Male Applicant’s.

And Update All the Address City to “Pune”, for Applicant those, who are eligible for Police Verification.

2] Write a batch class to show count of Account having Rating Hot or Warm.