39. Test Classes for Batch Classes - 24 Nov 2021

Batch Class:

global class BatchAccount implements Database.Batchable<sObject> {

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

return Database.getQueryLocator('SELECT Description,Name,Type FROM Account WHERE Type = \'Prospect\' or Type=\'Customer - Direct\' ');

}

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

if(!accountList.isEmpty()){

for(Account objAcc : accountList){

if(objAcc.Type == 'Prospect'){

objAcc.Description='Mere pass Bungla Hai, Gadi Hai, Bank Balance Hai, Tere Pass kya Hai?';

}

else{

if(objAcc.Type == 'Customer - Direct'){

objAcc.Description='Mere Pass Mask Hai';

}

}

}

Database.update(accountList,false);

}

}

global void finish(Database.BatchableContext BC){

}

}

Test Class:

@isTest

public class BatchAccountTest {

public static testMethod void prospectTestMethod(){

List<Account> accList = new List<Account>();

for(Integer i=1; i<=200; i++){

Account objAcc = new Account(Name='Cinemax'+i, Type='Prospect');

accList.add(objAcc);

}

if(!accList.isEmpty()){

Database.insert(accList, false);

}

Test.startTest();//To refresh the governor LIMIT

Database.executeBatch(new BatchAccount(), 200);

Test.stopTest();

List<Account> accListResult = [select id, Description from Account];

for(Account objAcc : accListResult){

System.assert(objAcc.Description.contains('Mere pass Bungla Hai, Gadi Hai, Bank Balance Hai, Tere Pass kya Hai?'));

}

}

public static testMethod void CustomerDirectTestMethod(){

List<Account> accList = new List<Account>();

for(Integer i=1; i<=200; i++){

Account objAcc = new Account(Name='Cinemax'+i, Type='Customer - Direct');

accList.add(objAcc);

}

if(!accList.isEmpty()){

Database.insert(accList, false);

}

Test.startTest();//To refresh the governor LIMIT

Database.executeBatch(new BatchAccount(), 200);

Test.stopTest();

List<Account> accListResult = [select id, Description from Account];

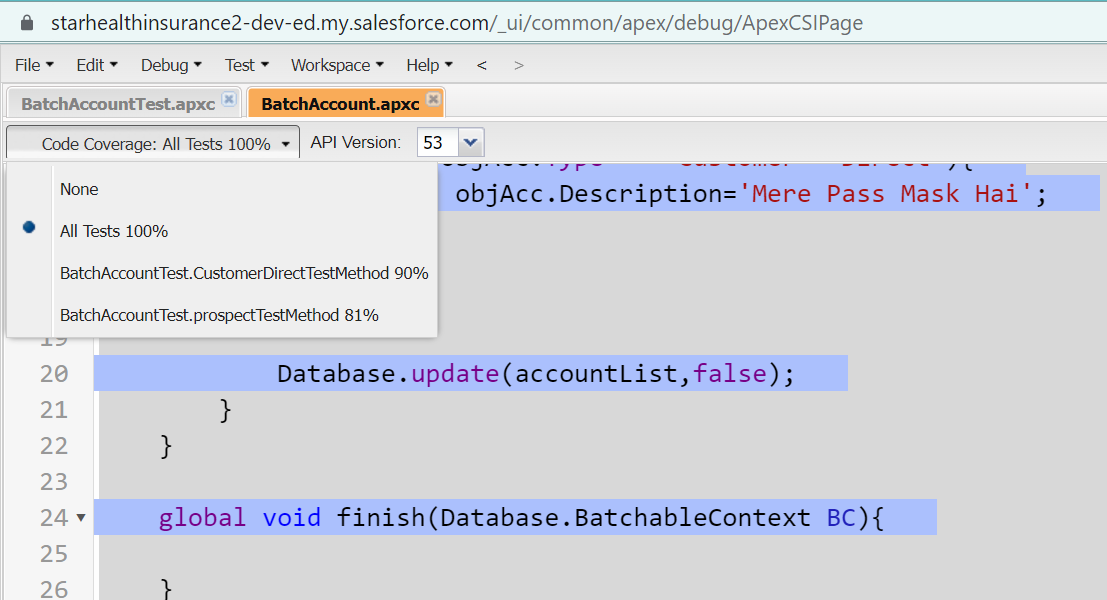
for(Account objAcc : accListResult){

System.assert(objAcc.Description.contains('Mere Pass Mask Hai'));

}

}

}



global class BatchContactAccount implements Database.Batchable<sObject>, Database.stateful {

integer count=0;

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

return Database.getQueryLocator('select id, FirstName, LastName, LeadSource, Account.Description from Contact');

}

global void execute(Database.BatchableContext BC, List<Contact> conList){//3

count++;//6

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

for(Contact objCon : conList){

accIdSet.add(objCon.AccountId);

}

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

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

accMap.put(objAcc.Id, objAcc);

}

List<Account> accListUpdate = new List<Account>();

for(Contact objCon : conList){

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

if(String.isNotBlank(accMap.get(objCon.AccountId).Description) || accMap.get(objCon.AccountId).Description !=null || accMap.get(objCon.AccountId).Description != ''){

accMap.get(objCon.AccountId).Description = accMap.get(objCon.AccountId).Description+ ' \n Contact '+objCon.FirstName+' '+objCon.LastName+' has Lead Source as Web';

}

else{

accMap.get(objCon.AccountId).Description = 'Contact '+objCon.FirstName+' '+objCon.LastName+' has Lead Source as Web';

}

}

}

accListUpdate.addAll(accMap.values());

Database.update(accListUpdate,false);

}

global void finish(Database.BatchableContext BC){

System.debug('#count = '+count);

}

}

@isTest

public class BatchContactAccountTest {

public static testMethod void testwithDescriptionMethod(){

Account objAcc = new Account(Name='Cinemax', Description=’Babu’);

insert objAcc;

List<Contact> conList = new List<Contact>();

for(Integer i=1; i<=200; i++){

Contact objCon = new Contact(FirstName='Bunty'+i, LastName='Modi'+i, AccountId=objAcc.Id);

conList.add(objCon);

}

if(!conList.isEmpty()){

Database.insert(conList,false);

}

Test.startTest();

Database.executeBatch(new BatchContactAccount(), 200);

Test.stopTest();

}

public static testMethod void testwithoutDescriptionMethod(){

Account objAcc = new Account(Name='Cinemax');

insert objAcc;

List<Contact> conList = new List<Contact>();

for(Integer i=1; i<=200; i++){

Contact objCon = new Contact(FirstName='Bunty'+i, LastName='Modi'+i, AccountId=objAcc.Id);

conList.add(objCon);

}

if(!conList.isEmpty()){

Database.insert(conList,false);

}

Test.startTest();

Database.executeBatch(new BatchContactAccount(), 200);

Test.stopTest();

}

}

global class BatchAccountContact implements Database.Batchable<sObject> {

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

return Database.getQueryLocator('select id, LeadSource,Account.Type from Contact where Account.Type= \'Prospect\' or Account.Type= \'Other\' ');

}

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

if(!conList.isEmpty()){

for(Contact objCon : conList){

if(objCon.Account.Type=='Prospect'){

objCon.LeadSource = 'Web';

}

else{

if(objCon.Account.Type=='Other'){

objCon.LeadSource = 'Phone Inquiry';

}

}

}

Database.update(conList,false);

}

}

global void finish(Database.BatchableContext BC){

}

}

@isTest

public class BatchAccountContactTest {

@isTest

public static void contactLeadSourceUpdate(){

Account objAcc = new Account(Name='Cinemax', Type='Prospect');

insert objAcc;

List<Contact> conList = new List<Contact>();

for(Integer i=0;i<200;i++){

Contact objCon = new Contact(FirstName='Alex'+i, LastName='Shah'+i, AccountId=objAcc.Id);

conList.add(objCon);

}

if(!conList.isEmpty()){

Database.insert(conList,false);

}

Test.startTest();

BatchAccountContact objBatchAccCon = new BatchAccountContact();

Database.executeBatch(objBatchAccCon,200);

Test.stopTest();

List<Contact> conListReturned = [select id, LeadSource from Contact where LeadSource='Web' and AccountId=:objAcc.Id];

System.assertEquals(200, conListReturned.size());

}

@isTest

public static void contactLeadSourceOtherUpdate(){

Account objAcc = new Account(Name='Cinemax', Type='Other');

insert objAcc;

List<Contact> conList = new List<Contact>();

for(Integer i=0;i<200;i++){

Contact objCon = new Contact(FirstName='Alex'+i, LastName='Shah'+i, AccountId=objAcc.Id);

conList.add(objCon);

}

if(!conList.isEmpty())

Database.insert(conList,false);

Test.startTest();

BatchAccountContact objBatchAccCon = new BatchAccountContact();

Database.executeBatch(objBatchAccCon,200);

Test.stopTest();

List<Contact> conListReturned = [select id, LeadSource from Contact where LeadSource='Phone Inquiry' and AccountId=:objAcc.Id];

System.assertEquals(200, conListReturned.size());

}

}

========================\\

VF Ka Update Wala:

public class AccountContactVF\_CX {

public List<Account> accList {get; set;}

public AccountContactVF\_CX(ApexPages.StandardController controller) {

accList = [select Name, (select FirstName,LastName,Salutation from Contacts) from Account where Name like 'P%' order by CreatedDate desc LIMIT 4];

}

public void updateRecords(){

if(!accList.isEmpty())

Database.update(accList,false); //Account Update

// Contact Update Start

List<Contact> conList = new List<Contact>();

for(Account objAcc : accList){

conList.addAll(objAcc.Contacts);

}

if(!conList.isEmpty())

Database.update(conList,false);

//Contact Update End

}

}

@isTest

public class AccountContactVF\_CXTest {

public static testMethod void testMethod1(){

Account objAcc = new Account(Name='Popcorn');

insert objAcc;

List<Contact> conList = new List<Contact>();

for(Integer i=0;i<200;i++){

Contact objCon = new Contact(FirstName='Alex'+i, LastName='Shah'+i, AccountId=objAcc.Id);

conList.add(objCon);

}

if(!conList.isEmpty()){

Database.insert(conList,false);

}

ApexPages.StandardController controller = new ApexPages.StandardController(objAcc);

AccountContactVF\_CX objAccountContactVF\_CX = new AccountContactVF\_CX(controller);

objAccountContactVF\_CX.updateRecords();

}

}