

## Batch Apex Class in Salesforce

### [1] What is batch apex class?

**Ans:** By using Batch apex classes we can process the records in batches in asynchronously. If you have a large number of records to process, for example, for data cleansing or archiving, batch Apex is your solution.

The execution logic of the batch class is called once for each batch of records. The default batch size is 200 records.

You can also specify a custom batch size. Furthermore, each batch execution is considered a discrete transaction.

With each new batch of records, a new set of governor limits is in effect. In this way, it's easier to ensure that your code stays within the governor execution limits.

Another benefit of discrete batch transactions is to allow for partial processing of a batch of records in case one batch fails to process successfully, all other batch transactions aren't affected and aren't rolled back if they were processed successfully.

### [2] Syntax of batch apex class

**Ans:**

```
global Class cleanup implements Database.Batchable<SObject> {

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

    global void execute(Database.BatchableContext BC,List){
    }

    global void Finish(Database.BatchableContext BC){
    }

}
```

### [3] What is Database.Batchable interface?

**Ans:** The class that implements this interface can be executed as a batch Apex job.

### [4] What are three methods of Database.Batchable<SObject> interface

**Ans:**

Start:

```
global Database.QueryLocator start (Database.BatchableContext bc){}
```

Execute:

```
global void execute(Database.BatchableContext bc,List <p>){}
```

Finish:

```
global void finish(Database.BatchableContext bc) {}
```

### [5] What is purpose of start method in batch apex?

**Ans:** It collect the records or objects to be passed to the interface method execute.

### [6] What is the Database.QueryLocator?

**Ans:** If we use a Database.QueryLocator, the governor limit for the total number of records retrieved by SOQL queries is bypassed. (Default 50,000 It allow up to 50 million records).

**[7] What is the iterable<Subject>?**

**Ans:** If you use an iterable, the governor limit for the total number of records retrieved by SOQL queries is still enforced.

**[8] What is the use of execute method?**

**Ans:** Contains or calls the main execution logic for the batch job.

**[9] How many times execute method is called?**

**Ans:** Execute method is called for each batch of records. (Number of total record / batch size).

**[10] What is the scope of execute method?**

**Ans:** The maximum value for the optional scope parameter is 2,000

**[11] What is the maximum size of the batch and minimum size of the batch ?**

**Ans:** The default batch size is 200 records. Min=01, max=2000

**[12] What is the Database.BatchableContext?**

**Ans:** BatchableContext Interface is Represents the parameter type of a batch job method and contains the batch job ID. This interface is implemented internally by Apex.

**[13] How to track the details of the current running Batch using BatchableContext?**

**Ans:** You can check the AsyncApexJob.Status using the JobId from the Database.BatchableContext.

**[14] How many batch jobs can be added to queue?**

**Ans:** Queued counts toward the limit of 5.

**[15] Where do you used batch apex class in your project**

**Ans:** Tell them the batch class implemented in you project.

OR

We have implemented a batch class to send email update to bulk Agents regarding to their commission on 01<sup>st</sup> Day of every month.

**[16] Difference between insert and Database.insert Or update and Database.update**

**Ans:**

**Insert**

Insert is a DML statement which is same as Database.insert to insert the record in database. If we use insert DML statement in bulk operation if error occurs in this operation, then execution stop and rollback all the insertion from the database those has been inserted successfully. When error occurs then you can handle this error by try catch block.

**Database.Insert**

If we use DML database method (Database.insert) to insert/update record in bulk, then if error occurs the remaining records will be inserted/updated means partially operation will be done.

Just take an Example:

If you are inserting 4 records.

2 having all the required fields value and remaining 2 records missing required fields value.

if you use

Database.insert(list) will throw an exception and it will not insert any one of the records.

But

if you use Database.insert(list, false) It will insert 2 records successfully and rest 2 records (required fields missing) will not be inserted.

Database.insert(list, false) In this scenario you will not get any exception

**[17] Suppose, finish method is blank, though it is important to write finish method**

**OR**

**If I don't want to write finish method, is it possible?**

**Ans :** Yes, we have to implement/write a finish method though it is blank. (Whenever we implement an interface, then we have to compulsorily implement all its method, so finish method is of Database.Batchable<Subject> interface).

**[18] How to execute a batch class**

**Ans:** BatchClassName obj = new BatchClassName();

Database.executeBatch(obj);

**OR**

Database.executeBatch(obj, 500);

**[19] What is use of Database.BatchableContext within the method**

**Ans:** Represents the parameter type of a batch job method and contains the batch job ID. This interface is implemented internally by Apex.

**[20] Can we call callouts from batch apex?**

**Ans:** Yes we can call.

**[21] Can we call another batch apex from batch apex?**

**Ans:** Yes you can call a batch apex from another batch apex .Either in start method or in finish method you can call other batch

**[22] How many callouts we can call in batch apex?**

**Ans:** Batch executions are limited to one callout per execution.

**[23] Batch is synchronous or Asynchronous operations?**

**Ans:** Asynchronous operations.

**[24] If I am inserting two record of the Account. I want a list of successful and failure records, how to do that.**

**Ans:**

```
// Create two accounts, one of which is missing a required field
```

```
Account[] accts = new List<Account>{  
    new Account(Name='Account1'),  
    new Account();  
};
```

```
Database.SaveResult[] srList = Database.insert(accts, false);
```

```
// Iterate through each returned result
```

```
for (Database.SaveResult sr : srList) {
```

```
    if (sr.isSuccess()) {
```

```
        // Operation was successful, so get the ID of the record that was processed
```

```
        System.debug('Successfully inserted account. Account ID: ' + sr.getId());
```

```
    }
```

```
    else {
```

```
        // Operation failed, so get all errors
```

```
        for(Database.Error err : sr.getErrors()) {
```

```
            System.debug('The following error has occurred.');
```

```
            System.debug(err.getStatusCode() + ': ' + err.getMessage());
```

```
            System.debug('Account fields that affected this error: ' + err.getFields());
```

```
        }
```

```
    }
```

```
}
```

**[25] What is batch chaining**

**Ans:** Calling one batch class from finish method of another batch class is known as Batch Chaining.

**[26] How many active batch classes can be there in queue.**

**Ans:** 05

**[27] Write a batch apex class to prefix Mr or Ms. as per Gender of the existing customer record (05 Lac). Show successful and failure customer Ids.**

**[28] How to schedule a batch class**

**Ans:** Write a schedulable class and from its execute method, call batch class

Example

ScheduleClass implements Schedulable{

```
    public void execute(SchedulableContext SC){  
        BatchClassName obj = new BatchClassName()  
        Database.executeBatch(obj);  
    }  
}
```

**[29] In how many ways can we schedule a class? Which are they?**

**Ans:**

Using Configuration: Go to Apex Class -> Schedule Apex Class

Using Customization : Using cron object:

```
ScheduleClass obj = new ScheduleClass ();  
String cronStr = '0 0,30 * * * *';  
System.schedule('Process Accs Job', cronStr, obj);
```

**[30] What is cron object and meaning of cron object expression?**

**Ans:** CRON expression we can schedule an apex with our customized date time or occurrence through code

```
0 0 5 ? * 1,2,3,4,5,6,7  
{1} {2} {3} {4} {5} {6}
```

{1} Seconds - so 0 here i.e. start of the minute.

{2} Minutes - 0 again so start of the hour.

{3} Hours - 5 so 5 am. Uses 24 hour notation so 21 = 9pm

{4} Day\_of\_month - ? means no specific value, only available for day of the month and day of the week.

{5} Month - \* indicates all values, i.e. every month. (if we only want to run on 1st Jan say, this would be 1)

{6} Day\_of\_week - 1,2,3,4,5,6,7 here specifies days 1,2,3,4,5,6,7 in the week. We could also write this string as MON-FRI or preferably as \* to indicate all values.

So this job reads to run at "0 seconds past 0 minutes of the 5th hour on no specified day of the month for every month of the year for every day of the week".

**[31] What will be con object expression, if I want to schedule an apex class on 15<sup>th</sup> May 2020 at 03:00 PM.**

**Ans :** String cronExp ='0 0 15 15 5 ? 2020'

**[20] What will be con object expression, if I want to schedule an apex class on 01<sup>st</sup> DAY of every month at 12:00 AM.**

**Ans:** String cronExp ='0 0 0 1 \* ? \*';

Expression	Description
0 0 0 ? ***	at 12:00 AM every day
0 0 12 ** ?	at 12:00 PM every day
0 0 10 ? **	at 10:00 AM every day
0 0 10 ** ?	at 10:00 AM every day
0 0 10 ** ? *	at 10:00 AM every day
0 0 15 ? ***	at 3:00 PM every day
0 0-5 15 ** ?	Every minute starting at 3:00 PM and ending at 3:05 PM, every day
0 15 17 ? * MON-FRI	at 5:15 PM every Monday, Tuesday, Wednesday, Thursday and Friday
0 25 5 15 * ?	at 5:25 AM on the 15th day of every month
0 15 17 ? * 6#3	at 5:15 PM on the third Friday of every month
0 0 17 ? * 6L	runs the last Friday of every month at 5:00 PM.
* 0 30 *** *;	every 30 minutes
0 0 23 ** ? 2018	runs every day at 11:00 PM during the year 2016.

**[32] How to Execute the Batch Apex Class for Every 30 minutes?**

Ans: `batchApexMethod exeClass = new batchApexMethod();`

`String cronStr = '0 0,30 * * * *';`

`System.schedule('Process Accs Job', cronStr, exeClass);`

**[33] From where to see scheduled apex job and delete a scheduled job**

Ans: Scheduled Jobs

**[34] What is AsyncApexJob class**

Ans: In Salesforce, records in the 'AsyncApexJob' table represents Apex jobs to run asynchronously when resources are available.

**[35] What is Database.State full interface?**

Ans: To maintain variable value inside the Batch class, Database.Stateful is used.

**[36] Is it possible to call future method from apex scheduler or not?**

Ans: Yes

```
//Scheduled Apex
public class DemoScheduler1 implements Schedulable{
    public void execute(SchedulableContext sc){
        system.debug('*****Going to call future method ');
        DemoAsynchronousTest.futureMethodCallFromScheduler();
    }
}

//apex class containing future method
public class DemoAsynchronousTest{
    @future
    public static void futureMethodCallFromScheduler(){
        system.debug('*****futureMethodCallFromScheduler get called');
    }
}
```

**[37] Is it possible to do Synchronous Web service callouts from scheduled apex?**

**Ans:** No.

Synchronous Web service callouts are not supported from scheduled Apex. To be able to make callouts, make an asynchronous callout by placing the callout in a method annotated with `@future(callout=true)` and call this method from scheduled Apex. However, if your scheduled Apex executes a batch job, callouts are supported from the batch class.

**[38] Can we call scheduler from future method?**

**Ans:** Yes

**[39] Is there is any way through which we can call future method from batch apex?**

**Ans:** As we know that a webservice can be called from batch class and webservice can call `@future` method. So in your batch class call webservice and which can call your `@future` method.

Also you can call future method from finish method in batch class.

**[40] What all different things which we need to consider while using future methods?**

**Ans:**

1. Methods with the future annotation cannot be used in Visualforce controllers in either `getMethodName` or `setMethodName` methods, nor in the constructor.
2. You cannot call a method annotated with future from a method that also has the future annotation. Nor can you call a trigger from an annotated method that calls another annotated method.
3. Future methods cannot be called from batch class.

**[41] Can we modify the scheduler class or classes referenced by this scheduler class if any scheduled job is pending?**

**Ans:** No

If there are one or more active scheduled jobs for an Apex class, you cannot update the class or any classes referenced by this class through the Salesforce user interface. However, you can enable deployments to update the class with active scheduled jobs by using the Metadata API.

**[42] What we can do if we have to deploy the scheduler class or classes referenced by this scheduler class if any scheduled job is pending for that scheduler class in target org?**

**Ans:** By default, changes to Apex code that have Apex jobs pending or in progress can't be deployed. To deploy these changes, do one of the following.

- Cancel Apex jobs before deploying changes to Apex code. Reschedule the jobs after the deployment.
- Enable deployments with Apex jobs in the Salesforce user interface in the Deployment Settings page.

**[43] What is Database.AllowCallouts?**

**Ans:**

- To use a callout in batch Apex, you must specify `Database.AllowsCallouts` in the class definition. For example:  

```
global class SearchAndReplace implements Database.Batchable<sObject>,  
Database.AllowsCallouts{  
    //Business logic you want by implementing Batchable interface methods  
}
```

Callouts include HTTP requests as well as methods defined with the `webservice` keyword.

**[44] When a BatchApexworker record is created?**

**Ans:**

- Ans: For each 10,000 `AsyncApexJob` records, Apex creates one additional `AsyncApexJob` record of type `BatchApexWorker` for internal use.