

POINTS TO REMEMBER

- use Database.getQueryLocator when you are using simple query to create scope for object in batch job.
- use Iterable<sObject> when you have complex criterion to process the record.
- If we use query locator object governor limit is bypassed.
- If we use Iterable<sObject> governor limit is enforced.
- The order of execution is not guaranteed in execute method.
- If the batch is executed without the optional scope parameter and If in batch apex suppose we have 600 jobs and there are total 200 records to be proceeds in each transaction,so in total we have 3 transaction.If first transaction succeeds but second fails so in this case the records update done in first transaction are not rolled back.

Number of batches and scope size:

When an apex job is executed by setting some value for scope size it is not necessary that the number of record processed in one batch iteration will be same as that of scope size specified.

Records from the query locator are retrieved in chunks of given chunk size called retrieveChunkSize .

Chunks size available are 100,200,400.

Selection of chunk size depends on the scope size specified.

If $1 \leq \text{scopeSize} \leq 100$, then $\text{retrieveChunkSize} = 100$

If $101 \leq \text{scopeSize} \leq 400$, then $\text{retrieveChunkSize} = 400$

If $401 \leq \text{scopeSize} \leq 2000$, then $\text{retrieveChunkSize} = 2000$

So Let say if we 285 records to be proceeds and we specify the scope size as 80 than the retrieveChunkSize will be 100.

In this case the total number of batches will note be 4 i.e (285/80) but in this case it would be 3 retrieve chunks and 6 execute chunks so the total number of batches will be 6.

1. Retrieve chunk 1: retrieve first 100 records

Execute chunk 1: pass first 80 records to execute() method

Execute chunk 2: pass remaining 20 records from this retrieve chunk to execute() method

2. Retrieve chunk 2: retrieve next 100 records

Execute chunk 3: pass first 80 records to execute() method

Execute chunk 4: pass remaining 20 records from this retrieve chunk to execute() method

3. Retrieve chunk 3: retrieve next 85 records

Execute chunk 5: pass first 80 records to execute() method

Execute chunk 6: pass remaining 5 records from this retrieve chunk to execute() method