

## Batch Apex:

Use:- When we want to process large no. of records - Asynchronously.

How:- Large records are broken into smaller batches

Types :- 1) Stateful Batch Apex:

Batch class maintains the State of each batch class execution.

↓  
This means that the ability to maintain Variable values b/w diff. batch executions

↳ This allows the batch class to maintain/keep information b/w transactions & resume processing if a failure occurs.

\* So, basically a state is maintained across transactions.

Example: If batch job is executing one logic & you need to send an Email at the end of batch job, with all all successful & failed records.

→ then, we can use Database.Stateful in class Definition.

\* SF allows 5 Batch Jobs to be Queued or Active Simultaneously.

If more jobs are submitted → they are held in "Flex Queue".



## \* Methods:

`Start()` → → Initializes Batch Job  
→ Returns the scope of records to be processed  
→ Uses Database Query Locator  
→ Called once at beginning of Batch job.

`Execute()` → Processes each Batch of records (Queried in start method)  
→ Called repeatedly for chunk of records &  
→ contains logic to perform the req op<sup>n</sup> on records.

`Finish()` → Called after all Batches have been processed.  
→ Used for Post-Processing tasks Ex ↓  
(i) Send email confirmation  
(ii) Initiating another batch job.

---

Database Query Locator: (Interface) (Used in Start method)

Used to efficiently fetch large datasets that needs to be processed.

---

\* → Each Batch has its own set of Governor limits & after one Batch is processed, a new set " " " are available.

---

## \* Best Practices of Batch Apex:

- 1) Use efficient SQL Query.
- 2) Handle exceptions properly.
- 3) Use smaller Batch sizes (if necessary)
- 4) Avoid complex logic in Execute Method
- 5) Ensure records are processed in a manner that don't exceed Governor limits