## When to use this spring batch

When u have large volume of data , and u wanted to do all at night / when u wanted to do all in single shot then go for batch processing

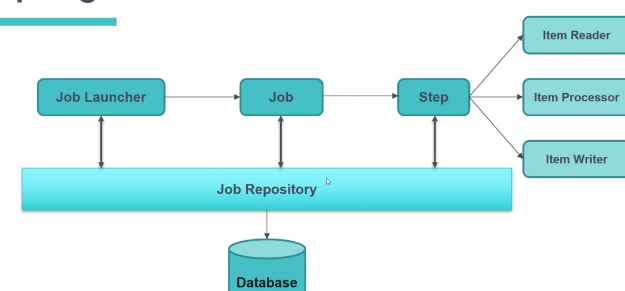
Ex:- sensex processing

### Features

We can skip the individual record which ever is failing with skip mechanism

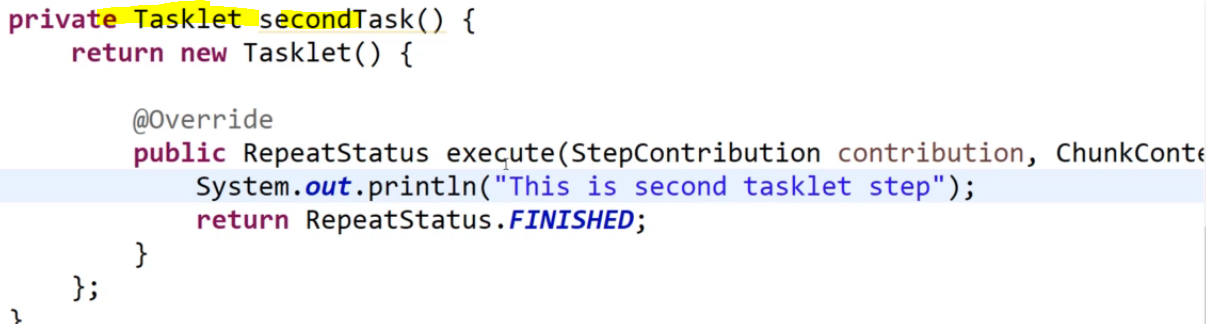


## Overview diagram

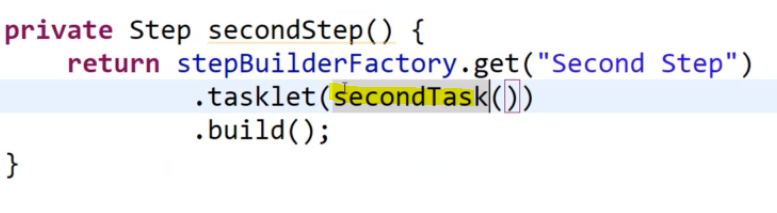


### Tasklet

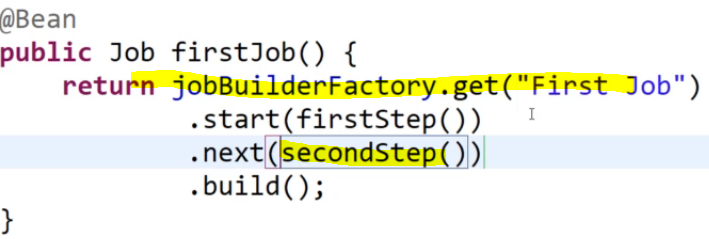
U can create the step with tasklet or u can also create a step with item Processor, reader, and writer



Configure the tasklet in the step

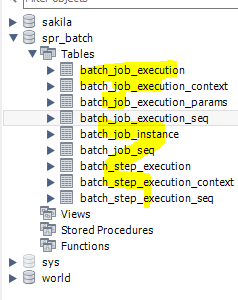


And then configure the step in that job



## **Tables created by spring batch**

These tables are created by spring batch by the file schema-oracle.sql file present in the spring batch jar



## JobInstance

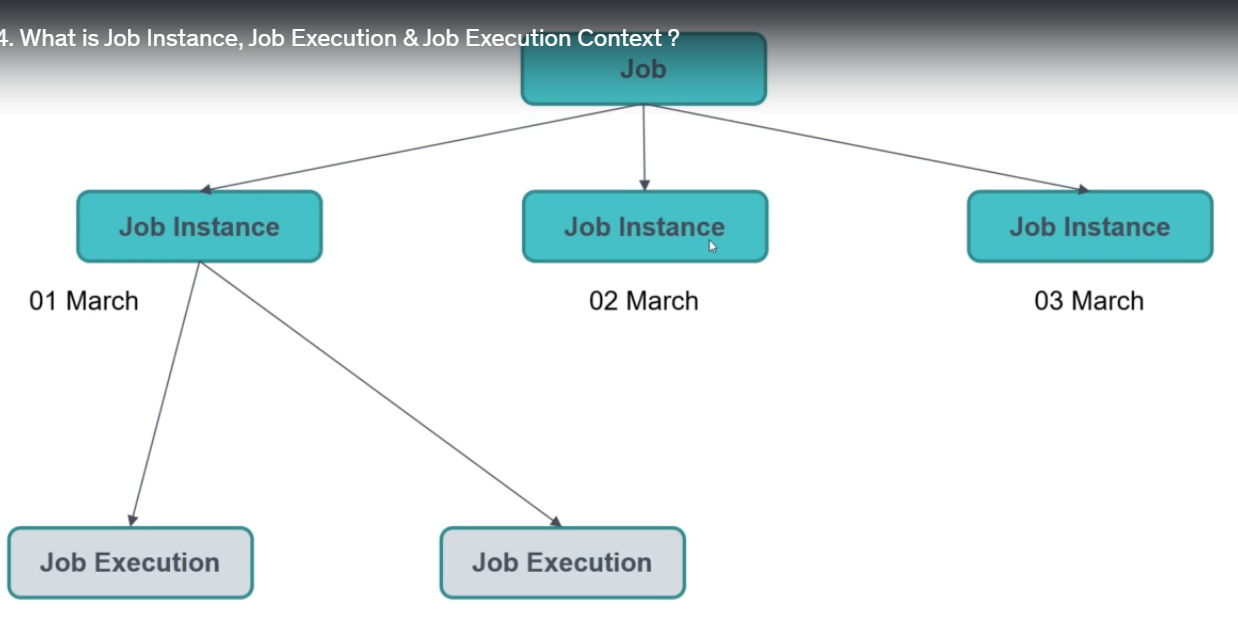
Its same like us, for 1 instance 1 time execution, for 1 person – 1 time marriage

This job instance status will be stored in table **batch\_job\_instance** refer **spring-batch-core jar for sql files** and job execution status will be stored in **batch\_job\_execution** table

Let’s say, every day the job will run, so there will be 1 job instance per day.

Under 1 job instance , we can have only 1 job execution, we cant have another job execution ,

But in the below diagram , since 1st job execution failed , only then we can have another job execution



Running the same job 2 / n times

##### solution -1 –with diff vm args

By passing the unique job parameters

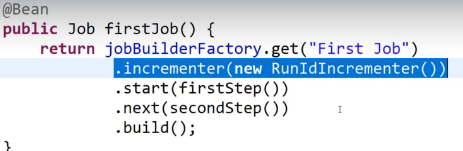
We can run the same job with diff job parameters in vm arguments

##### Solution -2 – use an incrementor

Instead of giving the vm arguments every time use below and configure the job

We should configure the incrementer to that job, else we should provide diff vm args each time

we can have multiple jobs in our appn



Script used by spring to create tables

Go to the jar and open that package an u will see the script used by spring to create tables

For dropping the table also we have the script , present in the same location

