Programa de Spring Batch

Spring Batch

Academia Java

Entregable

Natalia Esquivel Ocadiz

19 de septiembre 2025

Introducción:
Spring Batch es un framework Srping que permite crear sistemas robustos para procesamiento de grandes volúmenes de datos por lotes. En este proyecto se implementa Spring Batch para crear un sistema que procese datos de clientes de un csv (customers), los lea, los filtre y escriba en la base de datos Mysql. (sistema ETL, Extract, Transform, Load)

Objetivos:
- Crear un proyecto Spring Batch
- ItemProcessor filtre de todos los clientes solo aquellos que sean hombres, de China y que hayan nacido a partir del año 2000.

Implementación:
Primero, se agrega la dependencia de Spring Batch a nuestro pom.xml
Los componentes principales de la arquitectura Spring Batch son:
- JobLauncher: el cual es el que dispara el objeto Job
- Job: en él se define el trabajo a ejecutar, un job puede tener un step o múltiples steps:
    - Step: es la unidad de procesamiento del job, en el step se combinan de forma secuencial el ItemReader, ItemProcessor, ItemWriter:

  ItemReader: Se configuró un FlatFileItemReader para leer el archivo CSV de customers, mapeando cada línea a un objeto Customer mediante un LineMapper que convierte los campos del CSV a atributos del objeto.

  ItemProcessor: Se implementó un CustomItemProcessor que aplica la lógica de filtrado, evaluando tres condiciones: género masculino, nacionalidad china, y fecha de nacimiento posterior al año 2000. Solo los registros que cumplen los tres criterios pasan al siguiente paso.

  ItemWriter: Se configuró un JdbcBatchItemWriter que utiliza SQL preparado para insertar los registros filtrados en la base de datos MySQL de manera eficiente por lotes.

    -
- JobRepository: guarda toda la información importante del procesamiento de los datos

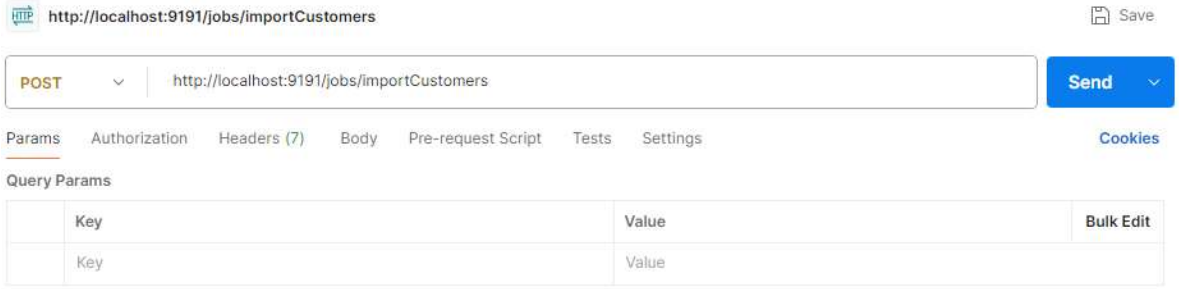
- Configuración de chunks y manejo de transacciones:

Se implementó el parámetro de chunk (es decir, procesará los datos en bloques de10),

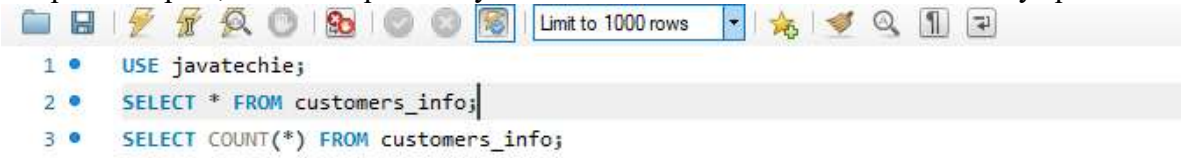
- Gestión de errores y reintentos

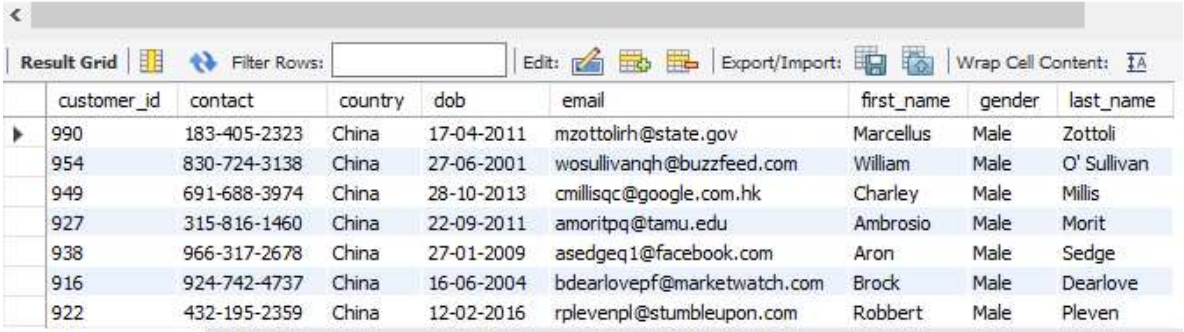
Se obtuvieron errores al querer filtrar el dato de Dob (date of birth), ya que este se encontraba como un String y no como un Date, por lo que se implementó un método anterior para transformar el tipo de dato de Dob a Date y de esta forma obtener el año (getYear()) y así usar la condicional >=2000.
Es importante levantar la base de datos del Workbench Mysql, para el ItemReader del Step.

Resultados:
Con el método Post se da inicio al JobLauncher



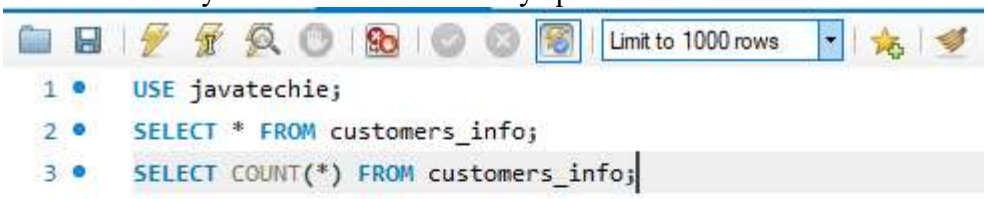
Después del post, se revisó que se hayan escrito los datos en la base de datos Mysql.

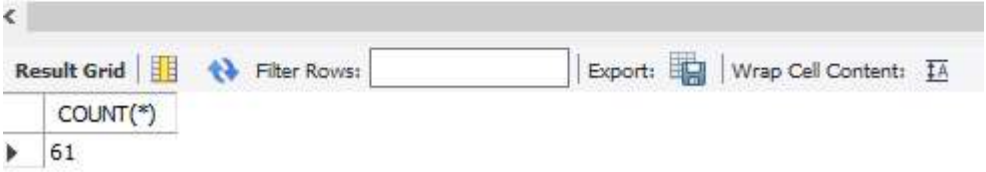




customer_id	contact	country	dob	email	first_name	gender	last_name
990	183-405-2323	China	17-04-2011	mzottolirh@state.gov	Marcellus	Male	Zottoli
954	830-724-3138	China	27-06-2001	wosullivanqh@buzzfeed.com	William	Male	O' Sullivan
949	691-688-3974	China	28-10-2013	cmillisqc@google.com.hk	Charley	Male	Millis
927	315-816-1460	China	22-09-2011	amoritpq@tamu.edu	Ambrosio	Male	Morit
938	966-317-2678	China	27-01-2009	asedgeq1@facebook.com	Aron	Male	Sedge
916	924-742-4737	China	16-06-2004	bdearlovepf@marketwatch.com	Brock	Male	Dearlove
922	432-195-2359	China	12-02-2016	rplevenpl@stumbleupon.com	Robbert	Male	Pleven

Datos filtrados y escritos en la BD de Mysql.





COUNT(*)
61

Datos procesados:

- Total de registros leídos: 1000 del archivo CSV
- Registros filtrados que cumplieron criterios: 61
- Registros escritos exitosamente en MySQL: 61

Performance del sistema:

- Tiempo total de ejecución: aproximadamente 6 segundos
- Uso de chunks de 10 registros optimizó el uso de memoria

Reflexiones:

Este proyecto demuestra cómo Spring Batch transforma el procesamiento de datos de una tarea manual y propensa a errores en un proceso automatizado, rápido y confiable.

Eficiencia lograda: En un procesamiento manual habría sido necesario insertar más de 1000 registros uno por uno aplicando filtros manualmente, lo cual sería lento y propenso a errores humanos. Con Spring Batch se logró un procesamiento automático de todo el dataset, aplicando filtros complejos (gender male, con nacionalidad china, y nacidos desde el 2000) de manera consistente y eficiente.

Arquitectura robusta: La separación clara de responsabilidades entre Reader, Processor y Writer facilita el mantenimiento y permite modificar cada componente independientemente. El manejo de transacciones por chunks garantiza la integridad de los datos.

Escalabilidad: El framework está preparado para manejar volúmenes de datos mucho mayores manteniendo un consumo de memoria controlado y tiempos de procesamiento predecibles.