JavaAcademy

# Spring Batch CSV to MySQL Processing

**Author**

Pedro Jahir Hinojosa García

**Instructor**

Miguel Rugerio

xideral®

**Delivery date: September 6, 2024**

# 1 Introduction

This document explains a Spring Batch program designed for a company that aims to identify potential stores as customers for selling dedicated internet services. The program reads data from a CSV file, filters relevant stores located in the city of Monterrey; because the database has initially all the citys in Nuevo León, and stores the data in a MySQL database. The database starts empty and is populated dynamically based on the request, targeting specific customers who match the company's criteria for offering dedicated internet.

# 2 Business Case

The company is focused on selling dedicated internet services to stores across various regions. To streamline their sales efforts, they need a system that can process a list of potential clients (stored in a CSV file), filter out irrelevant entries, and focus only on stores in Monterrey. These stores represent possible clients for dedicated internet services, making the system essential for efficiently identifying sales opportunities.

# 3 System Workflow

The Spring Batch program follows these steps:

- Read the CSV file containing customer data.
- Filter the data to only include stores located in Monterrey, which are potential customers for the company's dedicated internet service.
- Insert the filtered data into a MySQL table named `CUSTOMERS_INFO`.
- Use parallel processing to improve performance by handling records in chunks.

# 4 Java Components

The program consists of several components that work together to complete the batch job:

## 4.1 Main Application Class

The main application class initializes and starts the Spring Boot application:

```
@SpringBootApplication
public class BatchProcessingDemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(BatchProcessingDemoApplication.class, args);
    }
}
```

## 4.2 Batch Configuration

The batch configuration defines the job and steps needed to process the CSV file. Below is a summary of the important parts of the batch job:

- **CSV Reader**: Reads the data from the `Stores_Info.csv` file.
- **Item Processor**: Filters the data to only include stores in Monterrey, which are potential customers.
- **Item Writer**: Inserts the filtered data into the MySQL database.

## 4.3 Item Reader (CSV Reader)

The reader is responsible for reading the CSV file, skipping the header, and mapping each row to the `Customer` entity.

```
@Bean
public FlatFileItemReader<Customer> reader() {
    FlatFileItemReader<Customer> itemReader = new FlatFileItemReader<>();
    itemReader.setResource(new FileSystemResource("src/main/resources/Stores_Inf
    itemReader.setLinesToSkip(1);
    itemReader.setLineMapper(lineMapper());
    return itemReader;
}
```

## 4.4 Item Processor

The processor filters out records that do not meet the criteria (i.e., stores where the city is not Monterrey). It returns only records where the city is `Monterrey`.

```
.processor((ItemProcessor<Customer, Customer>) customer -> {
    if (customer.getCity().equals("Monterrey"))
        return customer;
    return null;
})
```

## 4.5 Item Writer

The writer is responsible for saving the processed data (filtered for Monterrey) into the `CUSTOMERS_INFO` table in the MySQL database.

```
Supplier<RepositoryItemWriter<Customer>> supplier = () -> {
    RepositoryItemWriter<Customer> writer = new RepositoryItemWriter<>();
    writer.setRepository(customerRepository);
    writer.setMethodName(”save”);
    return writer;
};
```

## 5 Database and Table Structure

The table in MySQL is named `CUSTOMERS_INFO`. It contains fields such as:

- `id`: Unique identifier for each customer.
- `comercial_name`: The commercial name of the store.
- `social_reason`: The legal name of the store.
- `activity`: The type of activity or business.
- `city`: The city in which the customer is located (only Monterrey in this case).
- `colony`: The colony (neighborhood) where the store is based.
- `postal_code`: The postal code for the store's location.
- `phone`, `email`, and other fields.

## 6 Output Example

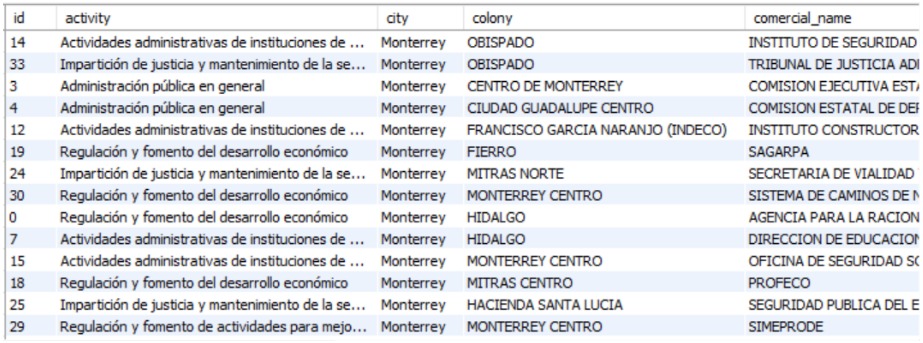
The following table shows the data inserted into the MySQL database after the batch job is executed. This output is filtered to include only stores in Monterrey, representing potential customers for the company's dedicated internet services.

As seen in Figure 1, the output contains the list of stores filtered by city, showing key information such as the store's commercial name, activity, colony, and city. This data is used by the company to identify sales opportunities for their dedicated internet services.

## 7 Controller

The `JobController` triggers the batch job through a REST API call. This endpoint runs the job and inserts the relevant records into the database.

```
@PostMapping(”/importCustomers”)
public void importCsvToDBJob() {
    JobParameters jobParameters = new JobParametersBuilder()
        .addLong(”startAt”, System.currentTimeMillis())
```



| id | activity | city | colony | comercial_name |
|---|---|---|---|---|
| 14 | Actividades administrativas de instituciones de ... | Monterrey | OBISPADO | INSTITUTO DE SEGURIDAD |
| 33 | Impartición de justicia y mantenimiento de la se... | Monterrey | OBISPADO | TRIBUNAL DE JUSTICIA AD |
| 3 | Administración pública en general | Monterrey | CENTRO DE MONTERREY | COMISION EJECUTIVA EST |
| 4 | Administración pública en general | Monterrey | CIUDAD GUADALUPE CENTRO | COMISION ESTATAL DE DE |
| 12 | Actividades administrativas de instituciones de ... | Monterrey | FRANCISCO GARCIA NARANJO (INDECO) | INSTITUTO CONSTRUCTOR |
| 19 | Regulación y fomento del desarrollo económico | Monterrey | FIERRO | SAGARPA |
| 24 | Impartición de justicia y mantenimiento de la se... | Monterrey | MITRAS NORTE | SECRETARIA DE VIALIDAD |
| 30 | Regulación y fomento del desarrollo económico | Monterrey | MONTERREY CENTRO | SISTEMA DE CAMINOS DE |
| 0 | Regulación y fomento del desarrollo económico | Monterrey | HIDALGO | AGENCIA PARA LA RACION |
| 7 | Actividades administrativas de instituciones de ... | Monterrey | HIDALGO | DIRECCION DE EDUCACIO |
| 15 | Actividades administrativas de instituciones de ... | Monterrey | MONTERREY CENTRO | OFICINA DE SEGURIDAD S |
| 18 | Regulación y fomento del desarrollo económico | Monterrey | MITRAS CENTRO | PROFECO |
| 25 | Impartición de justicia y mantenimiento de la se... | Monterrey | HACIENDA SANTA LUCIA | SEGURIDAD PUBLICA DEL E |
| 29 | Regulación y fomento de actividades para mejo... | Monterrey | MONTERREY CENTRO | SIMEPRODE |

Figure 1: Output of the batch job, showing stores in Monterrey as potential customers for dedicated internet services.

```
            .toJobParameters();
        try {
            jobLauncher.run(job, jobParameters);
        } catch (Exception e) {
            e.printStackTrace();
        }
}
```

## 8 Conclusion

This Spring Batch program efficiently processes a large CSV file, filters out unnecessary data, and inserts relevant customer information into a MySQL database. It is specifically designed for a company that sells dedicated internet services and needs to identify potential customers (stores) in Monterrey. The use of parallel processing and chunk-based processing makes the system scalable and capable of handling large data sets, helping the company streamline their sales efforts.