In the MyBatis Generator configuration, the **<table>** element is typically defined within the **generatorConfig.xml** file, not directly within the Maven plugin configuration in your **pom.xml** file. The **generatorConfig.xml** file provides a more comprehensive way to configure the code generation process, including specifying tables and their corresponding entity names.

Here's a step-by-step guide on how to create the **generatorConfig.xml** file and specify the table configuration:

1. **Create the generatorConfig.xml File**: In your project's **src/main/resources** directory, create a new file named **generatorConfig.xml**. This is where you'll define the details for table generation.
2. **Add Table Configuration**: Inside the **generatorConfig.xml** file, add the configuration for generating entities from your table:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN" "http://mybatis.org/dtd/mybatis-generator-config\_1\_0.dtd">

<generatorConfiguration>

<context id="default" targetRuntime="MyBatis3">

<!-- Other context configuration settings -->

<!-- Specify the table to generate -->

<table tableName="BrokerCalcuationClaims" domainObjectName="BCC" />

<!-- You can define additional tables here if needed -->

</context>

</generatorConfiguration>

Replace the placeholders as needed. The **<table>** element specifies the table name (**tableName**) and the desired entity name (**domainObjectName**) for code generation.

1. **Maven Plugin Configuration**: In your **pom.xml** file, configure the MyBatis Generator plugin to use your **generatorConfig.xml**:

<build>

<plugins>

<plugin>

<groupId>org.mybatis.generator</groupId>

<artifactId>mybatis-generator-maven-plugin</artifactId>

<version><!-- Use the latest version available --></version>

<configuration>

<configurationFile>src/main/resources/generatorConfig.xml</configurationFile>

</configuration>

<dependencies>

<!-- Add database-specific dependencies (SQL Server JDBC driver) -->

</dependencies>

</plugin>

</plugins>

</build>

1. **Run Code Generation**: Run the Maven command to generate code:

mvn mybatis-generator:generate

With this approach, you define your table and entity configuration in the **generatorConfig.xml** file, and the MyBatis Generator plugin reads that file to generate the entities accordingly. This provides a more organized and powerful way to configure the generation process for multiple tables and classes.

Please adapt this approach to your specific project structure and requirements, and ensure that the URLs and paths are correct in your configuration files.

<context id="default" targetRuntime="MyBatis3">

<!-- JDBC Connection -->

<jdbcConnection driverClass="com.microsoft.sqlserver.jdbc.SQLServerDriver"

connectionURL="jdbc:sqlserver://your-server:your-port;databaseName=your-database"

userId="your-username"

password="your-password">

</jdbcConnection>

<!-- Other configuration settings -->

<!-- Define tables to generate -->

<table tableName="your\_table\_name" domainObjectName="YourEntityName" />

</context>

<context ...>

...

<commentGenerator type="org.mybatis.generator.internal.DefaultCommentGenerator">

<property name="suppressAllComments" value="true"/>

</commentGenerator>

...

</context>