**Guide to Using Liquibase with MySQL**

**What is Liquibase?**

Liquibase is a database schema management tool that allows developers to define and manage database schema changes using declarative XML (or other formats like YAML, JSON, SQL).

**Why Use Liquibase?**

* Provides a version-controlled approach to database schema changes.
* Supports various database platforms including MySQL.
* Enables collaboration and consistency in database development.

**Getting Started with Liquibase**

Here’s an example of how you can use Liquibase with an XML changelog to manage changes to a MySQL database.

**Step 1: Setup Liquibase in Your Project**

If you are using Maven, add Liquibase dependencies to your pom.xml:

<dependency>

<groupId>org.liquibase</groupId>

<artifactId>liquibase-core</artifactId>

</dependency>

**Step 2: Configure Liquibase Properties**

Create a liquibase.properties file to configure the database connection:

# Liquibase properties

spring.liquibase.change-log=classpath:db/changelog/master.xml

spring.application.name=demo

#Server-Connection

server.port=8081

spring.datasource.url=jdbc:mysql://localhost:3306/demo

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

**Step 3: Create Master Changelog File**.

Create the master.xml file which will include other changelog files::

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

</databaseChangeLog>

**Step 4: Create an XML Changelog File**( 20240708194423\_created\_user\_entity.xml):

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

<changeSet id="97dca751-203f-48d4-9e8c-04d0096c9f16" author="logu">

<createTable tableName="user" >

<column name="id" type="INTEGER">

<constraints primaryKey="true" nullable="false"/>

</column>

<column name="name" type="VARCHAR(255)">

<constraints nullable="false"/>

</column>

<column name="address" type="VARCHAR(255)">

<constraints nullable="true"/>

</column>

<column name="number" type="INTEGER">

<constraints nullable="true"/>

</column>

</createTable>

</changeSet>

</databaseChangeLog>

**Explanation of the XML File:**

* databaseChangeLog: The root element that encapsulates all changes.
* changeSet: Represents a single change to be applied to the database.

It has an id attribute (unique within the changelog file) and an author attribute (typically your username or identifier).

* createTable: Specifies that a new table should be created.
* column: Defines columns within the table, specifying their name, type, and any constraints.
* constraints: Optional attributes that define rules for the column (e.g., primary key, nullable).

**Step 5: Added an XML Changelog File**(20240709110811\_added\_colum\_user\_entity.xml):

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

<changeSet id="ae7ab8b9-1e81-4d32-b9fb-b7ce09bff9ae" author="logu">

<addColumn tableName="user">

<column name="state" type="VARCHAR(255)">

<constraints nullable="true" />

</column>

</addColumn>

</changeSet>

</databaseChangeLog>

**Step 6: Updated an XML Changelog File(**20240709154022\_updated\_colum\_user\_entity.xml):

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

<changeSet id="b024ec5a-9165-4a5e-81f2-339ed6188a15" author="Logu">

<renameColumn tableName="user" oldColumnName="state"

newColumnName="district" columnDataType="VARCHAR(255)"/>

</changeSet>

</databaseChangeLog>

**Step 7: Dropped an XML Changelog File**

Dropped an XML changelog file (e.g., 20240709154022\_dropped\_user\_entity.xml) with the database changes:

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

<changeSet id="7b9456ca-0819-4aba-8157-93edcef6838a" author="Logu">

<dropColumn tableName="user" columnName="district"/>

</changeSet>

</databaseChangeLog>

**Step 8: Create Foreign key an XML Changelog File**(,20240709154022\_updated\_colum\_user\_entity.xml):

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

<changeSet id="b024ec5a-9165-4a5e-81f2-339ed6188a15" author="Logu">

<renameColumn tableName="user" oldColumnName="state" newColumnName="district" columnDataType="VARCHAR(255)"/>

</changeSet>

</databaseChangeLog>

**Step 9: Include Changelog Files in master.xml.**.

Use <include> tags to reference individual changelog files (\*.xml) that contain specific database changes

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

<databaseChangeLog

xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

<http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd>">

<include file="db/changelog/20240708194423\_created\_user\_entity.xml"/>

<include file="db/changelog/20240708202911\_created\_student\_entity.xml"/>

<include file="db/changelog/20240709123831\_created\_school\_entity.xml"/>

<includefile="db/changelog/20240709110811\_added\_colum\_user\_entity.xml"/>

<include

file="db/changelog/20240709154022\_updated\_colum\_user\_entity.xml"/>

<include file="db/changelog/20240709154022\_dropped\_user\_entity.xml"/>

<include

file="db/changelog/20240709125531\_added\_fk\_school\_details\_entity.xml"/>

</databaseChangeLog>

**Saving and Using master.xml**:

Save master.xml in your Liquibase project directory. Use Liquibase CLI to apply changes specified in master.xml to your MySQL database.

**Advanced Liquibase Features**

Using SQL Tag in Liquibase

The <sql> tag allows executing raw SQL statements within a changelog file, useful for operations not covered by Liquibase's built-in change types.

**Example: Adding a Stored Procedure**

**<changeSet id="2" author="your\_username">**

**<sql dbms="mysql">**

**<![CDATA[**

**DELIMITER //**

**CREATE PROCEDURE GetAllUsers()**

**BEGIN**

**SELECT \* FROM users;**

**END //**

**DELIMITER ;**

**]]>**

**</sql>**

**<rollback>**

**<sql dbms="mysql">**

**<![CDATA[**

**DROP PROCEDURE IF EXISTS GetAllUsers;**

**]]>**

**</sql>**

**</rollback>**

**</changeSet>**

**Parameterizing SQL Queries**

Liquibase supports parameterized SQL queries using <param> tags, allowing dynamic query execution.

**Example: Parameterized SQL Query**

**<changeSet id="3" author="your\_username">**

**<sql dbms="mysql">**

**<![CDATA[**

**SELECT username**

**FROM users**

**WHERE id = :userId**

**]]>**

**<param name="userId" value="1" dbmsType="int"/>**

**</sql>**

**</changeSet>**

**Conclusion**

By following this step-by-step guide, you should now have a solid understanding of how to use Liquibase with MySQL to manage database schema changes effectively using XML changelog files. Liquibase simplifies database management tasks, enhances collaboration, and ensures consistency across database environments.

**GITHUB Repository Link** :<https://github.com/Loganathan2300/Liquibase_poc>