**Liquibase**

Liquibase is an open-source database schema change management tool. It helps developers manage database changes (like creating, updating, or deleting tables) in a version-controlled manner, allowing them to track, version, and deploy database changes alongside application code. Key features include

You can write changes in different formats (like XML, YAML, JSON, or SQL formats.), and it works with various databases. It also lets you roll back changes if something goes wrong, making it easier to maintain your database alongside your application code.

**XML Format**

**1. Create Table**

<createTable tableName="person">

<column name="id" type="int" autoIncrement="true">

<constraints primaryKey="true"/>

</column>

<column name="name" type="varchar(255)"/>

</createTable>

**2. Add Column**

<addColumn tableName="person">

<column name="email" type="varchar(255)"/>

</addColumn>

**3. Drop Table**

<dropTable tableName="old\_table"/>

**4. Modify Column**

<modifyDataType tableName="person" columnName="name" newDataType="varchar(500)"/>

**5. Insert Data**

<insert tableName="person">

<column name="id" value="1"/>

<column name="name" value="John Doe"/>

</insert>

**6. Update Data**

<update tableName="person">

<column name="name" value="Jane Doe"/>

<where>id=1</where>

</update>

**7. Delete Data**

<delete tableName="person">

<where>id=1</where>

</delete>

**8. Drop Column**

<dropColumn tableName="person" columnName="email"/>

**9. Rename Column**

<renameColumn tableName="person" oldColumnName="name" newColumnName="full\_name" type="varchar(255)"/>

**10. Create Index**

<createIndex tableName="person" indexName="idx\_name">

<column name="name"/>

</createIndex>

**11. Drop Index**

<dropIndex indexName="idx\_name"/>

**Changelog**

A **changelog** in Liquibase is a file that contains a list of all the changes (called **changesets**) made to a database. It acts like a version control log for the database schema, helping to keep track of changes over time. Each changeset in the changelog file describes a specific operation to be performed on the database, such as creating a table, adding a column, or updating data.

Changelog files can be written in formats like **XML, YAML, JSON**, or **SQL**, and are used by Liquibase to apply the changes to the database in a consistent, versioned manner across different environments (development, testing, production). They also help in database rollback and reversion by maintaining a record of all changes.

In summary, a changelog is essential for database version control in projects using Liquibase, ensuring that all changes are tracked and can be applied or rolled back easily.

**1. Create Table**

<changeSet id="1" author="author\_name">

<createTable tableName="user">

<column name="id" type="int" autoIncrement="true">

<constraints primaryKey="true"/>

</column>

<column name="username" type="varchar(255)"/>

</createTable>

</changeSet>

**2. Add Column**

<changeSet id="2" author="author\_name">

<addColumn tableName="user">

<column name="email" type="varchar(255)"/>

</addColumn>

</changeSet>

**3. Insert Data**

<changeSet id="3" author="author\_name">

<insert tableName="user">

<column name="id" value="1"/>

<column name="username" value="admin"/>

<column name="email" value="admin@example.com"/>

</insert>

</changeSet>

**4. Update Data**

<changeSet id="4" author="author\_name">

<update tableName="user">

<column name="email" value="updated@example.com"/>

<where>id=1</where>

</update>

</changeSet>

**5. Drop Column**

<changeSet id="5" author="author\_name">

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

</changeSet>

</databaseChangeLog>