

LiquiBase Guidelines



SharepointEmpower
We collaborate your business world

Table of Contents

I.	What is LiquiBase?.....	2
II.	Why LiquiBase?.....	2
III.	Install LiquiBase on windows.....	3
IV.	LiquiBase Functionality.....	5
V.	LiquiBase Example?	6
VI.	How to run LiquiBase?	13
VII.	Reference Links.....	15



What is Liquibase?



- Liquibase is an open source, database independent library for tracking, managing and applying database changes.
- Database changes are stored in an XML file and (optionally) checked into source control.
- Liquibase executes changes based on this XML file to handle different revisions of database structures and data.

Why Liquibase?



- Consistent database changes
- Manage database at different states.
- Keep a history of changes.
- Automatic rollback support.
- Ability of automation.
- Effectively manage variable change
- Less human resources/errors.

Support different format



- XML
 - JSON
 - YAML
 - SQL
-

Install Liquibase on windows



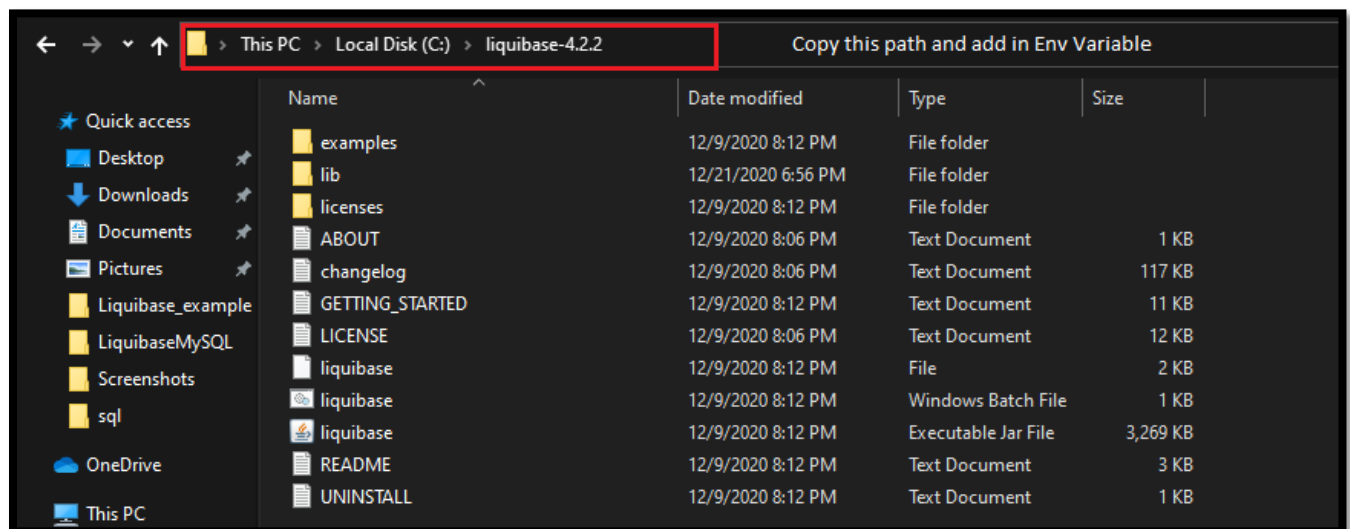
- Download Liquibase for Windows from below link.

<https://www.liquibase.org/download>

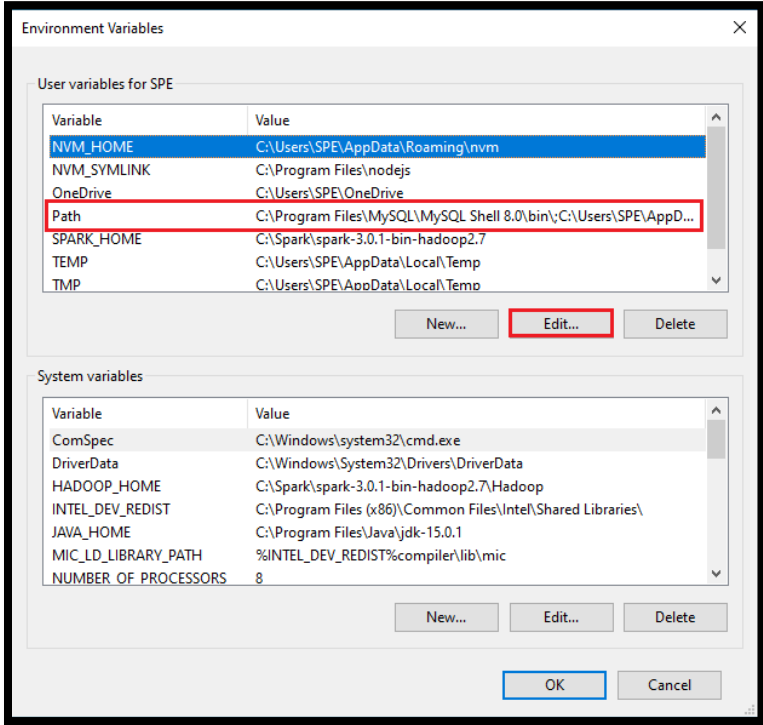
- Install Liquibase Desktop on Windows

1. Download Liquibase and extract in C drive and copy path and add in Environment User Variable.
2. Must Install Java In Local Machine.
3. Check Java Install or not Command: **Java -version**

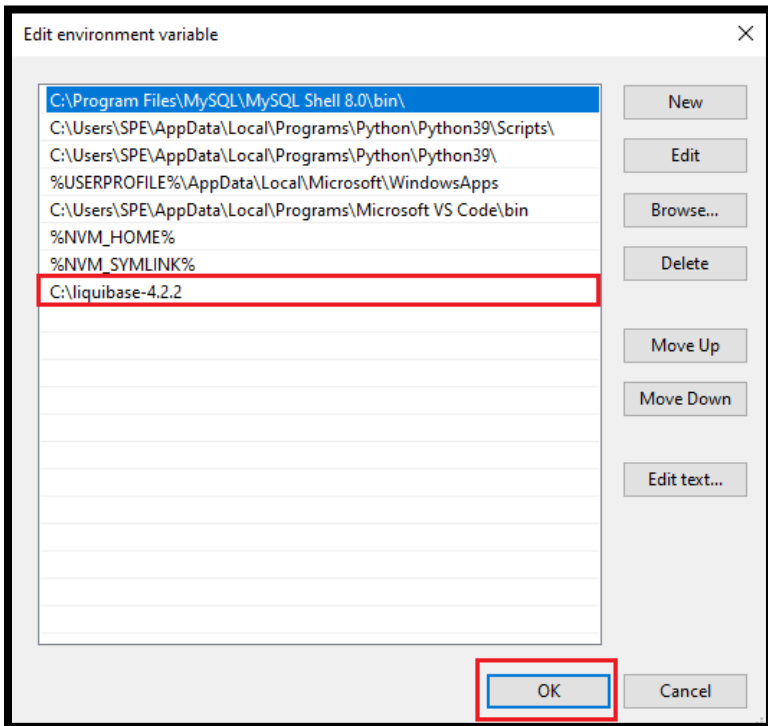
Step: 1



Step: 2



Step: 3



Liquibase Example?

Liquibase major concepts.

- Changelog file
- Changeset
- Changes

Liquibase changelog file

The root of all Liquibase changes is the database Changelog file.

```
<databaseChangeLog>
<changeSet id="1" ...>
...
</changeSet>
<changeSet id="2" ...>
...
</changeSet>
</databaseChangeLog>
```

It is possible to break up changelogs into multiple manageable pieces

```
<databaseChangeLog>
<include file="src/db/tables.xml"/>
<include file="src/db/views.xml"/>
<include file="src/db/data.xml"/>
...
</databaseChangeLog>
```



Liquibase changeset

Changeset is a group of changes.

Liquibase attempts to execute each changeset in a transaction that is committed at the end or rolled back if there is an error.

```
<databaseChangeLog>
<!-- id, author and current file name must be unique -->
<changeSet id="1" author="sfesenko" >
<comment>Sample of changeset</comment> <!-- comment is optional -->
<change .. /> <!-- will be explained later -->
<!-- rollback is a changeset that must be executed in order to "cancel" current changeset -->
<rollback>
<change .. />
</rollback>
</changeSet>
</databaseChangeLog>
```

```
<changeSet id="1" author="sfesenko" runAlways="true"> <!-- false by default -->
...
</changeset>
<changeSet id="2" author="sfesenko" runOnChange="true"> <!-- false by default -->
...
</changeset>
<changeSet id="3" author="sfesenko" failOnError="false"> <!-- true by default -->
...
</changeset>
```



Liquibase change

Each changeset contains change which describes required modification of database.

It is possible to use DSL (recommended, dbms independent) or SQL for defining a change.



Liquibase Example:

liquibase.properties

```
changeLogFile = dbchangelog.sql
url = jdbc:mysql://localhost:3306/myDatabase
username = root
password = Paras77@22
driver=com.mysql.jdbc.Driver
classpath = mysql-connector-java-5.1.49.jar
```

dbchangelog.sql

```
--liquibase formatted sql
```

```
create table PERSON (
  ID int not null,
  FNAME varchar(100) not null
);
```

```
--changeset Raj:40
insert into demo (id, name) values (10, 'joe');
insert into demo (id, name) values (20, 'John');
insert into demo (id, name) values (30, 'Tom');
```

```
-- DELETE Query
--changeset Bhautik:1
DELETE FROM demo WHERE id=10;
DELETE FROM demo WHERE id=20;
DELETE FROM demo WHERE id=30;
```

```
-- DROP TABLE Query
--changeset Bhautik:4
DROP TABLE person;
DROP TABLE test;
```

dbchangelog.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="1" author="bob">
        <createTable tableName="department">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false" />
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false" />
            </column>
            <column name="active" type="boolean" defaultValueBoolean="true" />
        </createTable>
    </changeSet>

    <changeSet id="2" author="Rahil">
        <createTable tableName="Test">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false" />
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false" />
            </column>
            <column name="active" type="boolean" defaultValueBoolean="true" />
        </createTable>
    </changeSet>

</databaseChangeLog>
```

To check your database, open your database IDE to find the change that you made.

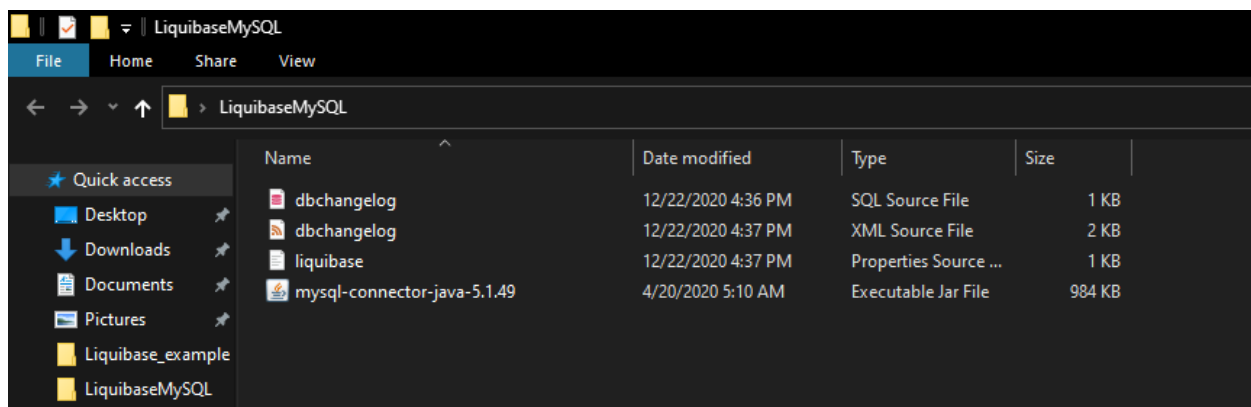
Notice that two tables were created along with test

- DATABASECHANGELOG
- DATABASECHANGELOGLOCK

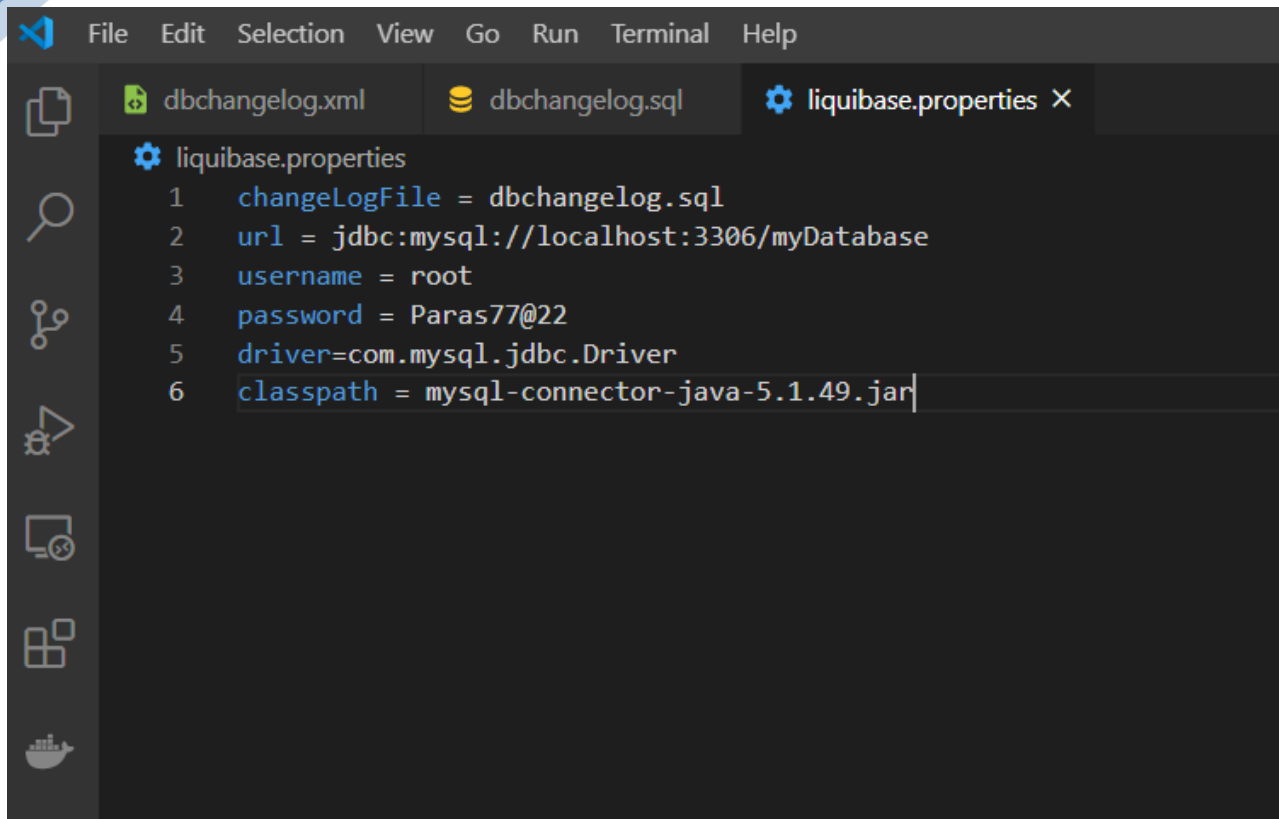
The **DATABASECHANGELOG** table contains a list of all the changes that have been run against the database.

The **DATABASECHANGELOGLOCK** table is used to make sure two machines don't attempt to modify the database at the same time.

Project Folder:



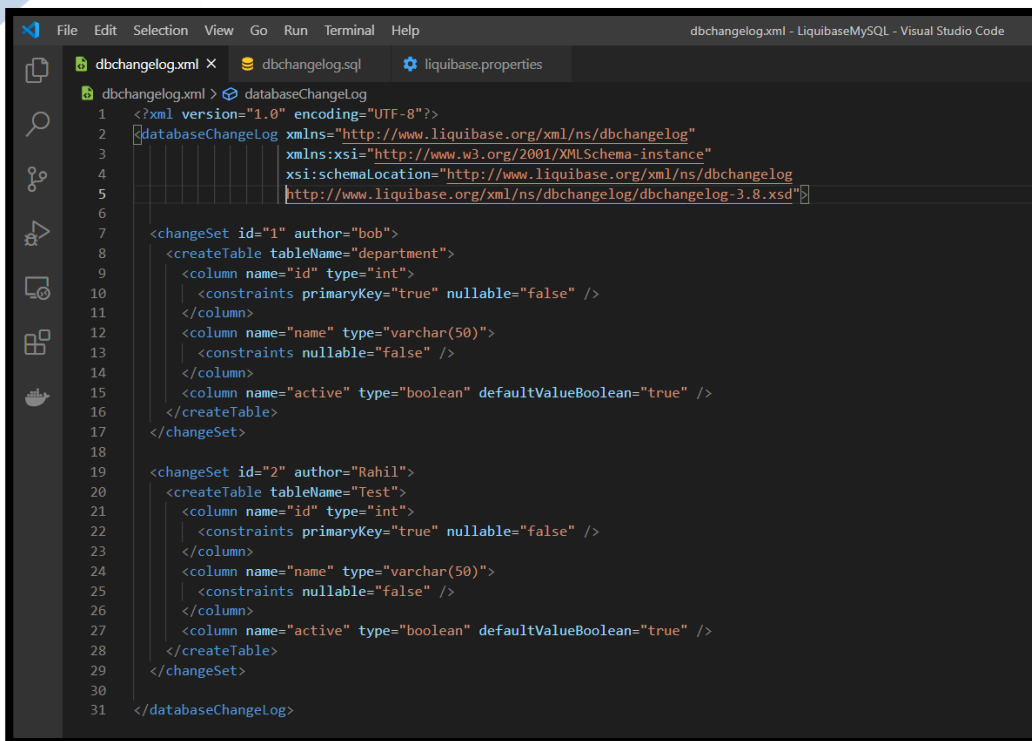
liquibase.properties:



The screenshot shows an IDE window with three tabs: `dbchangelog.xml`, `dbchangelog.sql`, and `liquibase.properties`. The `liquibase.properties` tab is active, displaying the following configuration:

```
1 changeLogFile = dbchangelog.sql
2 url = jdbc:mysql://localhost:3306/myDatabase
3 username = root
4 password = Paras77@22
5 driver=com.mysql.jdbc.Driver
6 classpath = mysql-connector-java-5.1.49.jar
```

dbchangelog.xml:

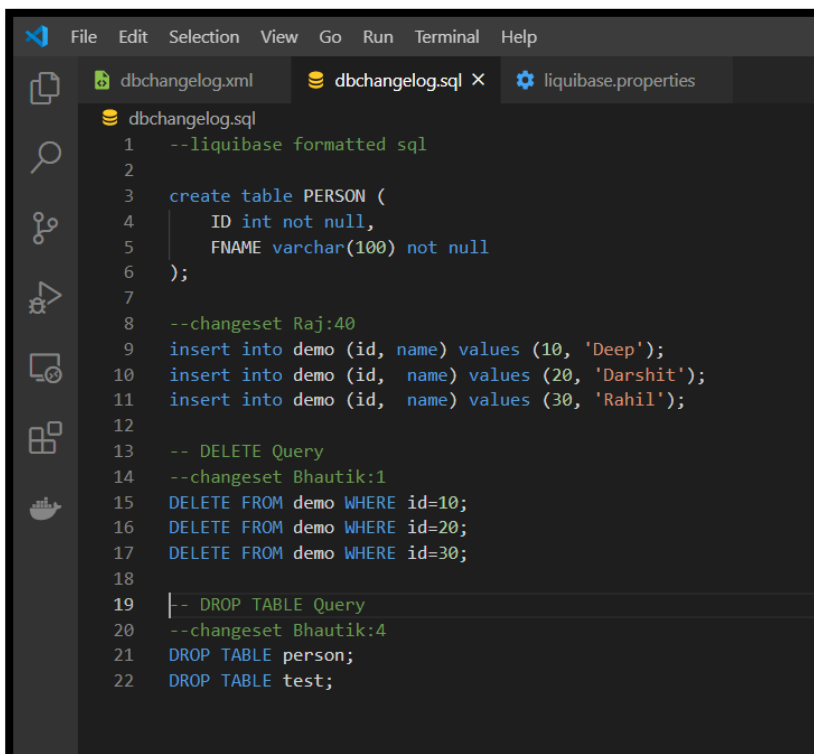


```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
3                      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4                      xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
5                      http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.8.xsd">
6
7      <changeSet id="1" author="bob">
8          <createTable tableName="department">
9              <column name="id" type="int">
10                 <constraints primaryKey="true" nullable="false" />
11             </column>
12             <column name="name" type="varchar(50)">
13                 <constraints nullable="false" />
14             </column>
15             <column name="active" type="boolean" defaultValueBoolean="true" />
16         </createTable>
17     </changeSet>
18
19     <changeSet id="2" author="Rahil">
20         <createTable tableName="Test">
21             <column name="id" type="int">
22                 <constraints primaryKey="true" nullable="false" />
23             </column>
24             <column name="name" type="varchar(50)">
25                 <constraints nullable="false" />
26             </column>
27             <column name="active" type="boolean" defaultValueBoolean="true" />
28         </createTable>
29     </changeSet>
30
31 </databaseChangeLog>

```

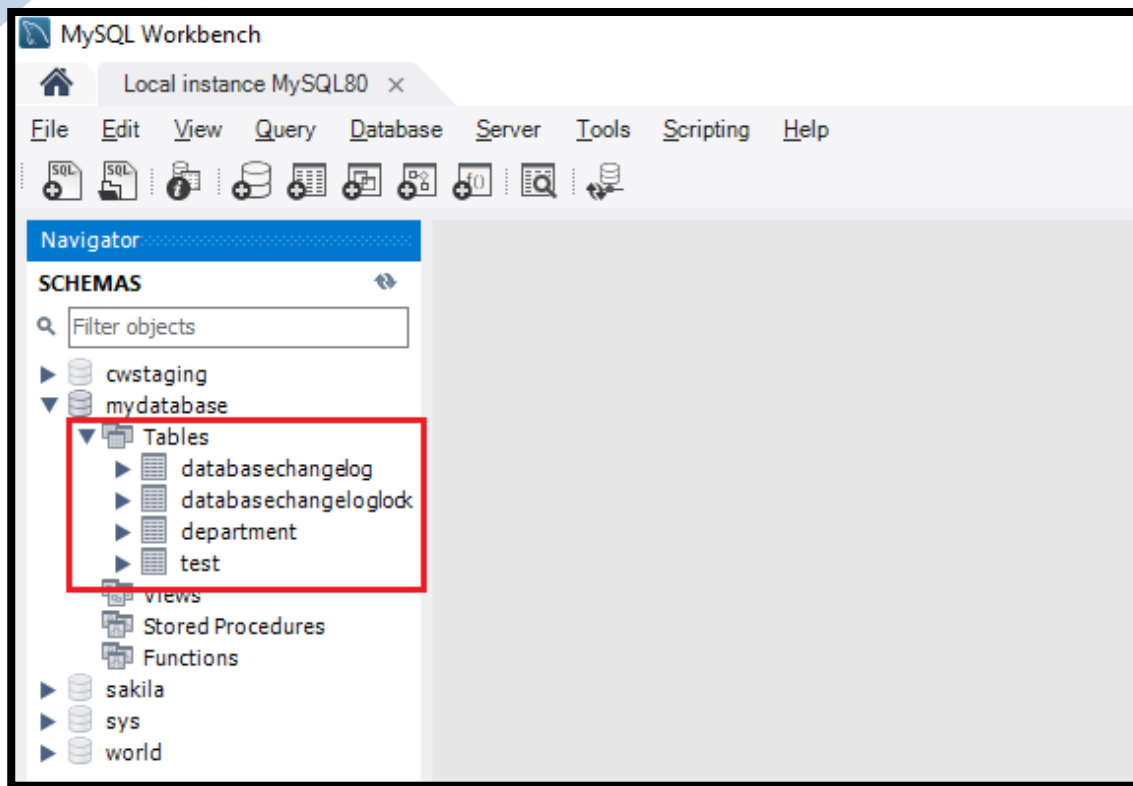
dbchangelog.sql:



```

1  --liquibase formatted sql
2
3  create table PERSON (
4      ID int not null,
5      FNAME varchar(100) not null
6  );
7
8  --changeset Raj:40
9  insert into demo (id, name) values (10, 'Deep');
10 insert into demo (id, name) values (20, 'Darshit');
11 insert into demo (id, name) values (30, 'Rahil');
12
13 -- DELETE Query
14 --changeset Bhautik:1
15 DELETE FROM demo WHERE id=10;
16 DELETE FROM demo WHERE id=20;
17 DELETE FROM demo WHERE id=30;
18
19 -- DROP TABLE Query
20 --changeset Bhautik:4
21 DROP TABLE person;
22 DROP TABLE test;

```

Now it will Create four Table in mydatabase

Reference Links



- Liquibase Database Tutorials : <https://docs.liquibase.com/workflows/database-setup-tutorials/home.html>
- Liquibase create Table Format Support:: <https://docs.liquibase.com/change-types/community/create-table.html>
- Liquibase Migrating with SQL: <https://docs.liquibase.com/workflows/liquibase-community/migrate-with-sql.html>
- Liquibase Community Commands: <https://docs.liquibase.com/commands/community/home.html>
- Liquibase Pro Commands: <https://docs.liquibase.com/commands/pro/home.html>
- Using Liquibase with MySQL: <https://docs.liquibase.com/workflows/database-setup-tutorials/mysql.html>
- Database migrations with multiple SQL files : <https://docs.liquibase.com/workflows/liquibase-community/multiple-sql-migration.html>
- Youtube Useful Link: <https://www.youtube.com/watch?v=8AdiGlb2SY4>
- Youtube Useful Link: <https://www.youtube.com/watch?v=HdXcf9E8ZVI>
- Some useful Command
 - liquibase update (direct run xml file command)
 - liquibase --changeLogFile=dbchangelog.sql update (**SQL File Run**)
 - liquibase --changeLogFile=dbchangelog.xml rollbackToDate 2020-12-22 11:18:23
 - liquibase rollbackOneUpdate --deploymentId=2126881174 -force
 - liquibase rollbackOneUpdateSQL --deploymentId=068379006 (**RollBack Command**)

Thank You

Bhautik Ladva
