



LIQUIBASE

Uma introdução

Marília Portela

CGE, 4 de Setembro de 2017

Liquibase em três frases

- Ferramenta de migração de banco de dados
- Aplicação stand-alone ou integrada ao projeto java
- Atualiza o esquema de qualquer SGBDR suportado

- Database.
- DB2.
- Derby.
- Firebird.
- H2.
- HyperSQL.
- Informix.
- MySQL.
- Oracle.
- PostgreSQL.
- SQL Server.
- SQLite.
- Sybase.
- Sybase Anywhere.

Changelog - o coração do liquibase

- Define operação principal do liquibase.
- Pode conter um ou mais changesets (unidades de mudança).
- Geralmente inclui outros changelogs, gerando uma hierarquia de arquivos

db.changelog-master.xml

```
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.8"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.8
    http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.8.xsd">

  <include file="db.changelog-1.0.xml" relativeToChangelogFile="true" />
  <include file="db.changelog-2.0.xml" relativeToChangelogFile="true" />

</databaseChangeLog>
```

Changelog - o coração do liquibase

- XML.
- SQL.
- JSON.
- YAML.
- Groovy.

```
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.8"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.8
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.8.xsd">

  <changeSet id="1" author="marilia">
    <addColumn catalogName="cat" schemaName="public" tableName="carro">
      <column name="modelo" type="varchar(255)"/>
    </addColumn>
  </changeSet>

  <changeSet id="2" author="bob">
    <renameColumn oldColumnName="modelo"
      newColumnName="tipo" tableName="carro" />
  </changeSet>

</databaseChangeLog>
```

Changelog - o coração do liquibase

- XML.
- **SQL**.
- JSON.
- YAML.
- Groovy.

```
--liquibase formatted sql
```

```
--changeset nvoxland:1
```

```
create table test1 (  
  id int primary key,  
  name varchar(255)  
);
```

```
--rollback drop table test1;
```

```
--changeset nvoxland:2
```

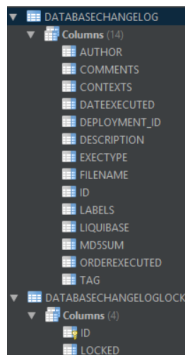
```
insert into test1 (id, name) values (1, 'name 1');  
insert into test1 (id, name) values (2, 'name 2');
```

```
--changeset nvoxland:3 dbms:oracle
```

```
create sequence seq_test;
```

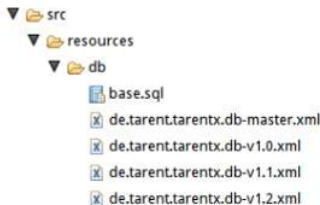
Aborgadem Geral

- Liquibase cria e mantém a tabela DATABASECHANGELOG que registra todas as mudanças executadas. Sempre que executa, ele a verifica para não reaplicar os mesmos changesets.
- A tabela DATABASECHANGELOCK também é criada e gerenciada pelo Liquibase para fins de controle de concorrência.



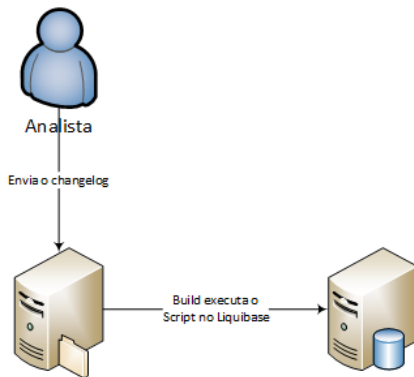
Aborgadem Geral

- Mudanças descritas de forma estritamente incremental.
- Quando uma mudança (changeset) é introduzida no changelog (e potencialmente executada no BD), **ela não deve ser alterada.**



Execução

- Gradle.
- Maven.
- Software CI.
- **ServletListener.**
- **Linha de comando com a aplicação stand-alone.**



Execução

Configuração ServletListener

Arquivo web.xml

```
<context-param>
  <param-name>liquibase.changelog</param-name>
  <param-value>/liquibase/changelog/db.changelog-master.xml</param-value>
</context-param>

<context-param>
  <param-name>liquibase.datasource</param-name>
  <param-value>java:/liquibaseDS</param-value>
</context-param>

<context-param>
  <param-name>liquibase.host.includes</param-name>
  <param-value>
    localhost,
    e-parceriashomol.cge.local,
    e-parceriasdesenv.cge.local,
    e-parceriashomol2.cge.local
  </param-value>
</context-param>

<listener>
  <listener-class>liquibase.integration.servlet.LiquibaseServletListener</listener-class>
</listener>
```

*Já que o usuário do banco de dados configurado para o liquibase precisa ter permissões de criação de objetos de BD, é aconselhável criar um datasource (liquibaseDS) diferente do usado pela aplicação.

Execução

Configuração Linha de Comando

liquibase.properties

```
changeLogFile=changelog-master.xml  
driver: org.hsqldb.jdbc.JDBC.Driver  
classpath: D:/Tools/HSQldb/hsqldb-2.3.1/hsqldb/lib/hsqldb.jar  
url: jdbc:hsqldb:hsq://localhost/teste-liquibase-s  
username: root  
password: root
```

*Note os caminhos são relativos ao diretório atual.

**Liquibase procura o arquivo liquibase.properties no diretório atual, ou no caminho provido na opção `-defaultsFile` do comando liquibase

Execução

Usando a Linha de Comando

Criando tag e voltando ao ponto da tag

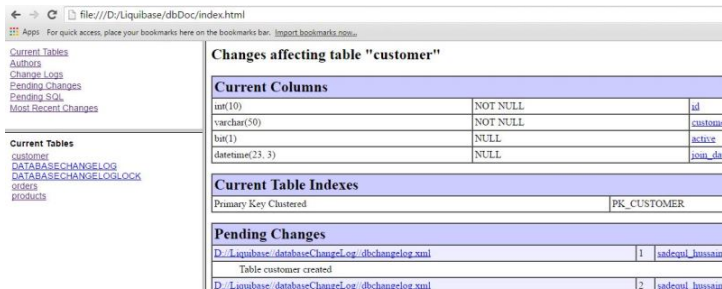
```
liquibase tag "tag1"
```

```
liquibase update
```

```
liquibase rollback "tag1"
```

Gerando documentação sobre as mudanças

```
liquibase dbdoc documentacao
```



The screenshot shows the Liquibase web interface in a browser. The address bar displays 'file:///D:/Liquibase/dbDoc/index.html'. The left sidebar contains a navigation menu with links: 'Current Tables', 'Authors', 'Change Logs', 'Pending Changes', 'Pending SQL', and 'Most Recent Changes'. Below this, under 'Current Tables', are links for 'customer', 'DATABASECHANGELOG', 'DATABASECHANGELOGLOCK', 'orders', and 'products'. The main content area is titled 'Changes affecting table "customer"'. It contains three sections: 'Current Columns', 'Current Table Indexes', and 'Pending Changes'. The 'Current Columns' section is a table with 3 columns: data type, constraints, and name. The 'Current Table Indexes' section shows a primary key 'PK_CUSTOMER'. The 'Pending Changes' section is a table with 3 columns: change ID, description, and author.

Column Name	Data Type	Constraints	Comments
id	int(10)	NOT NULL	
customer	varchar(50)	NOT NULL	
active	bit(1)	NULL	
join_date	datetime(23, 3)	NULL	

Index Name	Index Type	Index Columns
PK_CUSTOMER	Primary Key Clustered	id

Change ID	Description	Author
1	Table customer created	sadequi_hussain
2		sadequi_hussain

Refatorações(DDLs)

Criação de Tabelas

```
<changeSet id="1" author="marilia">

  <createTable tableName="tabelaA">
    <column name="id" type="bigint" autoIncrement="true">
      <constraints primaryKey="true" nullable="false"/>
    </column>
    <column name="codigo" type="varchar(20)">
      <constraints nullable="false" unique="true"/>
    </column>
    <column name="descricao" type="varchar(200)"/>
  </createTable>

  <addUniqueConstraint columnNames="codigo,descricao"
    constraintName="codigo_desc" tableName="tabelaA"/>

  <createTable tableName="tabelaB">
    <column name="id" type="bigint" autoIncrement="true">
      <constraints primaryKey="true" nullable="false"/>
    </column>
    <column name="id_tabelaA" type="bigint"/>
  </createTable>

  <addForeignKeyConstraint baseColumnNames="id_tabelaA" baseTableName="tabelaB"
    referencedColumnNames="id" referencedTableName="tabelaA" constraintName="id_tabelaA_fk"/>

</changeSet>
```

Refatorações(DDLs)

O liquibase é capaz de gerar o rollback do changeset automaticamente, com exceção de alguns casos especiais, nos quais ele deve ser explicitamente descrito.

Rollbacks

```
<databaseChangeLog
xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.8"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.8
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.8.xsd">

  <changeSet id="1" author="marilia">
    <dropTable tableName="nomeTabela"/>
    <rollback>
      <createTable tableName="nomeTabela">
        <column name="id" type="int"/>
      </createTable>
    </rollback>
  </changeSet>
</databaseChangeLog>
```

Refatorações(DDLs)

Criação de Views

```
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.8"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.8
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.8.xsd">

  <changeSet id="1" author="marilia">
    <createView catalogName="cat" schemaName="public" viewName="vi_pessoa"
      replaceIfExists="true">
      SELECT id, name FROM pessoa WHERE id > 10
    </createView>
  </changeSet>

</databaseChangeLog>
```

Refatorações(DDLs)

Criação de Procedures e Triggers

```
<changeSet id="1" author="marilia">
  <createProcedure comments="Um comentário" dbms="postgreSQL">
    CREATE OR REPLACE FUNCTION change_update_time() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
    BEGIN
    NEW.updated_at := CURRENT_TIMESTAMP;
    RETURN NEW;
    END;
    $$;
  </createProcedure>
  <rollback>
    DROP FUNCTION change_update_time();
  </rollback>
</changeSet>

<changeSet id="2" author="marilia">
  <sql dbms="postgreSQL">
    DROP TRIGGER IF EXISTS trigger_auditoria ON tabelaA;
    CREATE TRIGGER trigger_auditoria
    BEFORE UPDATE ON tabelaA
    FOR EACH ROW EXECUTE PROCEDURE change_update_time();
  </sql>
  <rollback>
    DROP TRIGGER trigger_auditoria ON tabelaA;
  </rollback>
</changeSet>
```

Refatorações(DDLs)

Pré-condições

```
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.8"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.8
http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.8.xsd">
  <preConditions>
    <dbms type="oracle" />
    <runningAs username="SYSTEM" />
  </preConditions>

  <changeSet id="1" author="marilia">
    <preConditions onFail="WARN">
      <sqlCheck expectedResult="0">select count(*) from tabelaVelha</sqlCheck>
    </preConditions>
    <comment>
      Comentários vêm depois da pré-condição.
      Do contrário, liquibase mostra um erro.
      'onFail' e 'onError' admitem os atributos: HALT(default), CONTINUE, MARK_RUN e WARN
    </comment>
    <dropTable tableName="tabelaVelha"/>
  </changeSet>
</databaseChangeLog>
```


Carregando Dados

Insert

```
<changeSet id="2" author="marilia">
  <insert schemaName="nome-esquema" tableName="tabelaB">
    <column name="descricao">0 registro foi finalizado</column>
    <column name="id_tabelaA">
      (SELECT id FROM tabelaA WHERE codigo='FINALIZADO')
    </column>
  </insert>
</changeSet>
```

Carregando Dados

Load

```
<changeSet id="2" author="marilia">  
  <loadData file="db.changelogs/dados.csv"  
    schemaName="nome_esquema" quotchar="'" tableName="tabelaA"/>  
</changeSet>
```

dados.csv

Codigo,Descricao

'EFETIVADO','Item finalizado'

'AUTORIZADO','Item autorizado'

'CANCELADO','Item Cancelado'

Mais Informações

Links:

Melhores práticas

Syntax nos changelogs

Atualização

Linha de comando

DEMONSTRAÇÃO