

Liquibase Tutorial - Praktický kurz (60 minut)

1. Úvod do Liquibase (10 minut)

Co je Liquibase?

- Nástroj pro verzování databázového schématu
- Umožňuje sledování a správu databázových změn
- Podpora různých databázových systémů (PostgreSQL, MySQL, Oracle, atd.)
- Integrace s různými build nástroji (Maven, Gradle)

Proč používat Liquibase?

- Verzování DB schématu společně s kódem
- Automatizace nasazení DB změn
- Podpora rollbacků
- Různé formáty zápisu změn (XML, YAML, JSON, SQL)
- Multiplatformní podpora
- Test-driven development pro databáze

2. Základní koncepty (15 minut)

Changelog

- Hlavní soubor popisující všechny změny
- Hierarchická struktura
- Možnost rozdělení do více souborů

```
<?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-4.9.xsd">

  <include file="changes/v1.0/01-create-tables.xml"/>
  <include file="changes/v1.0/02-insert-data.xml"/>
</databaseChangeLog>
```

Changeset

- Základní jednotka změny
- Unikátní identifikace (id + author)
- Neměnnost po nasazení
- Možnost rollbacku

```
<changeSet id="1" author="developer">
  <createTable tableName="users">
    <column name="id" type="bigint" autoIncrement="true">
      <constraints primaryKey="true" nullable="false"/>
    </column>
    <column name="username" type="varchar(50)">
      <constraints nullable="false" unique="true"/>
    </column>
  </createTable>
</changeSet>
```

Preconditions

- Podmínky pro provedení změn
- Kontrola stavu databáze
- Řešení závislostí

```
<changeSet id="2" author="developer">
  <preConditions onFail="MARK_RAN">
    <not>
      <tableExists tableName="users"/>
    </not>
  </preConditions>
  <!-- changeset content -->
</changeSet>
```

3. Praktické příklady (25 minut)

Příklad 1: Základní struktura projektu

```
src/
main/
resources/
db/
changelog/
db.changelog-master.xml
changes/
v1.0/
01-create-tables.xml
02-insert-data.xml
```

Příklad 2: Vytvoření schématu

```

<!-- 01-create-tables.xml -->
<changeSet id="1" author="developer">
  <!-- Products table -->
  <createTable tableName="products">
    <column name="id" type="bigint" autoIncrement="true">
      <constraints primaryKey="true" nullable="false"/>
    </column>
    <column name="name" type="varchar(100)">
      <constraints nullable="false"/>
    </column>
    <column name="price" type="decimal(10,2)">
      <constraints nullable="false"/>
    </column>
    <column name="created_at" type="timestamp" defaultValueComputed="CURRENT_TIMESTAMP"/>
  </createTable>

  <!-- Orders table -->
  <createTable tableName="orders">
    <column name="id" type="bigint" autoIncrement="true">
      <constraints primaryKey="true" nullable="false"/>
    </column>
    <column name="order_date" type="timestamp" defaultValueComputed="CURRENT_TIMESTAMP"/>
    <column name="status" type="varchar(20)" defaultValue="NEW">
      <constraints nullable="false"/>
    </column>
  </createTable>

  <!-- Foreign keys and indexes -->
  <createIndex indexName="idx_product_name"
    tableName="products">
    <column name="name"/>
  </createIndex>
</changeSet>

```

Příklad 3: Modifikace schématu a data

```

<!-- 02-modify-schema.xml -->
<changeSet id="2" author="developer">
  <!-- Add new column -->
  <addColumn tableName="products">
    <column name="category" type="varchar(50)" />
  </addColumn>

  <!-- Update existing data -->
  <update tableName="products">
    <column name="category" value="GENERAL" />
    <where>category is null</where>
  </update>

  <!-- Add not-null constraint -->
  <addNotNullConstraint
    tableName="products"
    columnName="category"
    defaultNullValue="GENERAL" />
</changeSet>

```

Příklad 4: Rollback

```

<changeSet id="3" author="developer">
  <createTable tableName="customers">
    <column name="id" type="bigint" autoIncrement="true">
      <constraints primaryKey="true" nullable="false" />
    </column>
    <column name="name" type="varchar(100)" />
  </createTable>

  <rollback>
    <dropTable tableName="customers" />
  </rollback>
</changeSet>

```

4. Best Practices a Tipy (10 minut)

Organizace změn

1. Struktura souborů

- Logické členění do adresářů podle verzí
- Jeden changeset = jedna logická změna
- Konzistentní pojmenování souborů

2. Verzování

- Použití sémantického verzování
- Tagování důležitých verzí
- Neměnit již nasazené changesety

3. Rollback

- Definovat rollback kde je to možné
- Testovat rollback procedury
- Dokumentovat nemožnost rollbacku

Běžné příkazy

```
# Update databáze
./gradlew update

# Generování SQL
./gradlew updateSQL

# Status změn
./gradlew status

# Rollback poslední změny
./gradlew rollbackCount -PliquibaseRollbackCount=1

# Označení verze tagem
./gradlew tag -PliquibaseCommandValue=v1.0

# Rollback na tag
./gradlew rollback -PliquibaseCommandValue=v1.0
```

Bezpečnost a výkon

1. Bezpečnost

- Správa přístupových údajů
- Řízení oprávnění
- Audit změn

2. Výkon

- Indexy pro často používané dotazy
- Postupné nasazování velkých změn
- Testování na reprezentativních datech

Integrace s CI/CD

1. Automatizace

- Součást deployment pipeline
- Automatické testy
- Kontrola změn před nasazením

2. Prostředí

- Použití kontextů pro různá prostředí
- Správa konfigurací
- Monitorování nasazení

Cvičení pro samostatnou práci

Cvičení 1: Základní schéma

Vytvořte changeset pro:

- Tabulku customers s poli id, name, email
- Unikátní index na email
- Vložení testovacích dat

Cvičení 2: Modifikace schématu

Upravte schéma:

- Přidejte sloupec phone_number do customers
- Vytvořte tabulku customer_addresses
- Definujte foreign key vztah
- Připravte rollback scénář

Cvičení 3: Pokročilé funkce

Implementujte:

- Preconditions pro kontrolu existence tabulek
- Tagy pro označení verze
- Kontexty pro různá prostředí
- Custom SQL pro složitější změny

Další zdroje

1. Dokumentace

- [Oficiální Liquibase dokumentace \(https://docs.liquibase.com\)](https://docs.liquibase.com)
- [Best Practices Guide \(https://www.liquibase.org/get-started/best-practices\)](https://www.liquibase.org/get-started/best-practices)
- [Liquibase Blog \(https://www.liquibase.com/blog\)](https://www.liquibase.com/blog)

2. Nástroje

- Liquibase CLI
- Gradle plugin
- Maven plugin
- IDE integrace

3. Komunita

- GitHub repository
- Stack Overflow tag
- Liquibase fórum