

DB séma kezelés Liquibase-el

Németh Zoltán
Backend lead, Ustream

A probléma

- gyakran változó, nagy db sémák
- sok fejlesztő, saját local db a fejlesztéshez
-> hibaforrás (eltérő db-re fejleszt mint az éles)
- ahhoz hogy működjön localban a site kellenek a db-be adatok is
- automatizált DB tesztek futtatásához is egységes db kell

schema.sql fileok a repoban

- első megoldási kísérlet, így mindenkinek hozzáférhető legalább a séma
- nem futtatható inkrementálisan, nem alter-ek hanem mindig a pillanatnyi állapot leképezése, mint egy dump
- nincs rollback
- teljesen manuális, azon múlik minden hogy
 - rendszeren updatelik-e
 - rendszeren lefuttatják-e
- részben automatizálható, pl mysqldump

Liquibase

*"Liquibase is an open source (Apache 2.0 Licensed), database-independent library for tracking, managing and applying database changes. It is built on a simple premise: **All database changes are stored in a human readable yet trackable form and checked into source control.**"*

<http://www.liquibase.org/>

Fő jellemzők

- Java executable
- többféleképp futtatható: pl Ant, Maven, command line
- XML changelogok
- Update
- Rollback
- Generate changelog

Changelog

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<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-2.0.xsd">

  <changeSet id="default-values-000" author="fqqdk" runAlways="true">
    <!-- Default character set is utf8 -->
    <sql>SET character_set_server=utf8</sql>
    <rollback></rollback>
  </changeSet>

  <include file="sql/liquibase/init/centralSchemaTables.xml" />
  <include file="sql/liquibase/init/centralSchemaFunctions.xml" />
  <include file="sql/liquibase/init/centralSchemaTriggers.xml" />
  <include file="sql/liquibase/init/centralSchemaViews.xml" />

  <includeAll path="sql/liquibase/changelogs/" />
</databaseChangeLog>
```

Changelog

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<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-2.0.xsd">
  <changeSet author="fqqdk (generated)" id="initial-import-tables-0">
    <sql>SET FOREIGN_KEY_CHECKS=0</sql>
  </changeSet>
  <changeSet author="fqqdk (generated)" id="initial-import-tables-1">
    <sql><![CDATA[
CREATE TABLE `acl` (
  `requester_id` int(10) unsigned NOT NULL COMMENT 'A hozzaferest kero entitas azonositoja, pl. "add_ha
  `subject_id` int(10) unsigned NOT NULL COMMENT 'A hozzaferes targya, pl. "add_hashtag_video" eseten a
  `usecase` varchar(50) collate utf8_unicode_ci NOT NULL COMMENT 'A hasznalati eset, pl. "add_hashtag_v
  `access_type` enum('deny','allow') collate utf8_unicode_ci NOT NULL default 'deny',
  PRIMARY KEY (`requester_id`,`subject_id`,`usecase`,`access_type`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
]]>
    </sql>
  </changeSet>
  <changeSet author="fqqdk (generated)" id="initial-import-tables-2">
    <sql><![CDATA[
CREATE TABLE `ad_free_unlimited_plan` (
```

Changelog (alter xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<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-2.0.xsd">
  <changeSet author="medvetalp" id="channel_subscription_enabled_add_status">
    <addColumn tableName="channel_subscription_enabled">
      <column name="status" type="ENUM('free','paid')" remarks="Current subscription-status of channel." >
        <constraints nullable="false"/>
      </column>
    </addColumn>
  </changeSet>
</databaseChangeLog>
```


XML tagekkel leírható műveletek

- add column, remove column
- modify column, rename column

- create / rename / drop table
- create / rename / drop view

- constraint műveletek (not null, unique, foreign key)

- auto increment, default value

- create / drop index

- adat kezelés (insert, load, stb)

Changelog (alter sql)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<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-2.0.xsd">
  <changeSet author="znemeth" id="alter_ppv_transactions">
    <sql><![CDATA[
      ALTER TABLE `ppv_transactions` ADD `purchase_location` VARCHAR(255) COLLATE utf8_unicode_ci NULL DEFAULT NULL
    ]]></sql>
    <rollback><![CDATA[
      ALTER TABLE `ppv_transactions` DROP COLUMN `purchase_location`;
    ]]></rollback>
  </changeSet>
</databaseChangeLog>
```

Futtatás

```
liquibase --changeLogFile  
sql/liquibase/<changelog>.xml --url  
jdbc:mysql://localhost/<adatbazisnev>  
<parancs>
```

- fontosabb parancsok:
 - update
 - rollback
 - dry run: updateSql, rollbackSql
 - changeset szám alapján: updateCount, rollbackCount
 - tag

Update

- Changeset azonosítás id, author, filename alapján
- DATABASECHANGELOG nevű táblában tárolja a már lefutott changeseteket
- az update parancs hatására lefuttatja azokat amik nincsenek még a táblában

```
mysql> select * from DATABASECHANGELOG;
```

ID	AUTHOR	FILENAME	DATEEXECUTED	ORDEREXECUTED	EXECTYPE	MD5SUM	DESCRIPTION
default-values-000	fqqdk	sql/liquibase/centralSchemaMaster.xml	2012-01-13 15:39:11	1	EXECUTED	3:ee31f6bd013f8f4d1e6cf9f313b6a5f3	Custom SQL
initial-import-tables-0	fqqdk (generated)	sql/liquibase/init/centralSchemaTables.xml	2012-01-13 15:39:11	2	EXECUTED	3:cc124077cda1cfb0c70c1ec823551223	Custom SQL

Rollback

- ha xml tagekkel adtuk meg az altert, vannak módosítás típusok amiknél adott a rollback módja (pl addColumn)
- ha custom sql-el adjuk meg (illetve ha nem egyértelmű a rollback módja pl drop), meg kell adnunk rollback sql-t is
- tagelni lehet a db állapotát
- rollbacknek meg lehet adni hogy melyik tagre menjen vissza vagy hány changesetet

Bevezetés

- eredeti elképzelés: folyamatos automatikus generálás
 - initial changelog az éles db alapján
 - periodikusan futtatva diff megállapítása
 - ebből diff changelog generálása
- végül ezt elvetettük mert
 - changelog generálás nem volt túl gyors
 - diff generálás időnként hibázott illetve elhalt
 - rollback statement-ek generálása nem nagyon ment

Választott megoldás

- initial changelog generálás (végül a repoban levő schema.sql alapján, majd később az éles db alapján kiegészítettük)
- változások manuális managementje:
 - changelog xml-ek kézzel íródnak
 - commitoljuk őket
 - az éles db-t nem automatikusan változtatjuk (ez a backend review miatt is fontos)
- tesztekhez használt local db updatelését bekötöttük a test bootstrap-be

Mit sikerült megoldani

- minden fejlesztőnek rendelkezésre áll a központival megegyező db
- a szükséges kiinduló dataset is megvan
- egységes a db tesztkörnyezet, a CI környezet számára is
- viszonylag egyszerű a változás kezelése a korábbiakhoz képest (schema.sql kézi módosíthatása)
- minden adatbázisunk le van fedve liquibase-el

Mit szeretnénk még megoldani

- Legyen legenerálható bármely db teljes séma sql-je (pl ha új gépet húzunk fel). Ezt több módon is el lehet képzelni:
 - liquibase-el felhúzzuk egy local db-be majd onnan mysqldump
 - vagy a fapados megoldás, egy script kiszedi az xml-ekből az sql-eket - viszont akkor a tag-es changesetek kezelése lesz macerás
- Minél több dolgot xml tagekkel írjunk le custom sql helyett
- Legyen mindenhez rollback statement ;)

Kérdések

Köszönöm a figyelmet

syntaxerror@syntaxerror.hu