12.05.2015
Tallinn

LiquiBase vs Flyway

Baltic DevOps

Andrei Solntsev
//codeborne
well-crafted software

Agile

Change -

it

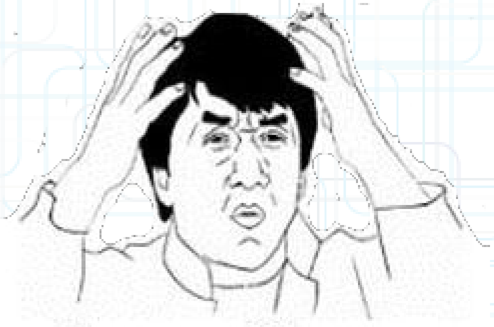
should be

easy

Database

Change is complex

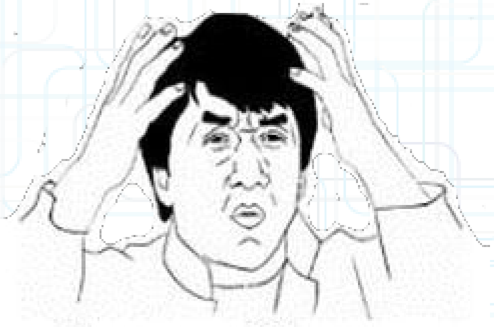
- Tables
- Live data!



Database

Change is complex

- Tables
- Live data!
- Stored procedures
- Pl/Sql packages
- Materialized views
- Triggers
- DB Links

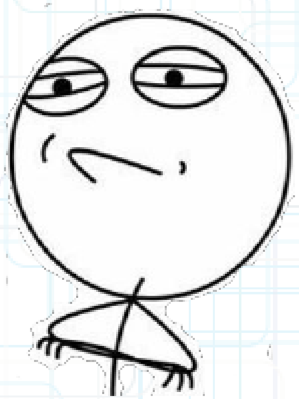


Life BEFORE liquibase

```
CREATE TABLE PERSON (
    first_name VARCHAR2(16),
    last_name VARCHAR2(16)
);
```

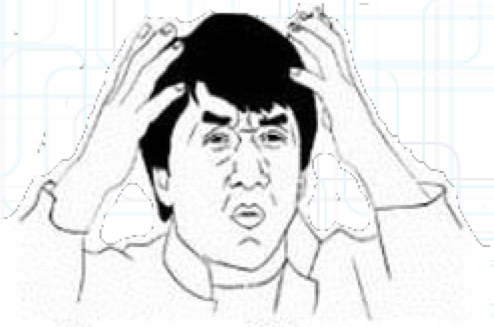
Life BEFORE liquibase

```
CREATE TABLE PERSON (
    first_name VARCHAR2(16),
    middle_name VARCHAR2(2),
    last_name VARCHAR2(16)
);
```



Life BEFORE liquibase

```
CREATE TABLE PERSON (
    first_name VARCHAR2(32),
    middle_name VARCHAR2(32),
    last_name VARCHAR2(32)
);
```



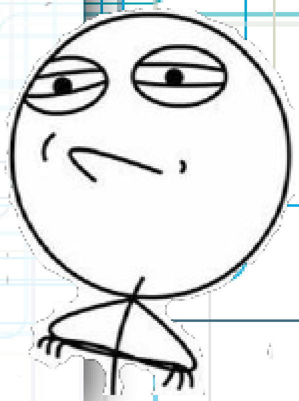
Solution:
small steps

1.sql CREATE TABLE PERSON (first_name, last_name)

2.sql ALTER TABLE PERSON ADD COLUMN middle_name

3.sql ALTER TABLE PERSON DROP COLUMN middle_name

That's evolution!



Tools

- Flyway DB
- LIQUIBASE

- simple
- powerful

Ruby ON Rails

http://rubydev.ru

south

DBDeploy

play

Flyway: just files

V1__blah.sql

V2_blah.sql

V3_blah.sql

LiquiBase: XML

```
<sqlFile path="1.sql"/>


<sqlFile path="2.sql"/>


<sqlFile path="3.sql"/>
```

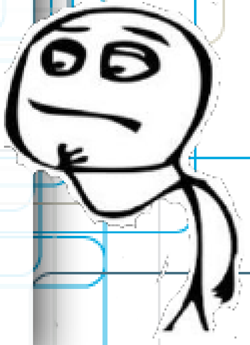
`changelog.xml`

LiquiBase: XML

```
<changeSet id="1">
    <sqlFile path="1.sql"/>
</changeSet>

<changeSet id="2">
    <sqlFile path="2.sql"/>
</changeSet>

<changeSet id="3">
    <sqlFile path="3.sql"/>
</changeSet>
```

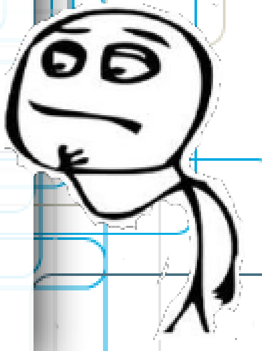
LiquiBase: XML

```
<changeSet id="1">
    <sqlFile path="1.sql"/>
</changeSet>

<changeSet id="2" runAlways="true">
    <sqlFile path="2.sql"/>
</changeSet>

<changeSet id="3">
    <sqlFile path="3.sql"/>
</changeSet>
```



LiquiBase: XML

```
<changeSet id="1">
    <sqlFile path="1.sql"/>
</changeSet>

<changeSet id="2" runAlways="true">
    <sqlFile path="2.sql"/>
</changeSet>

<changeSet id="3"runOnChange="true">
    <sqlFile path="3.sql"/>
</changeSet>
```



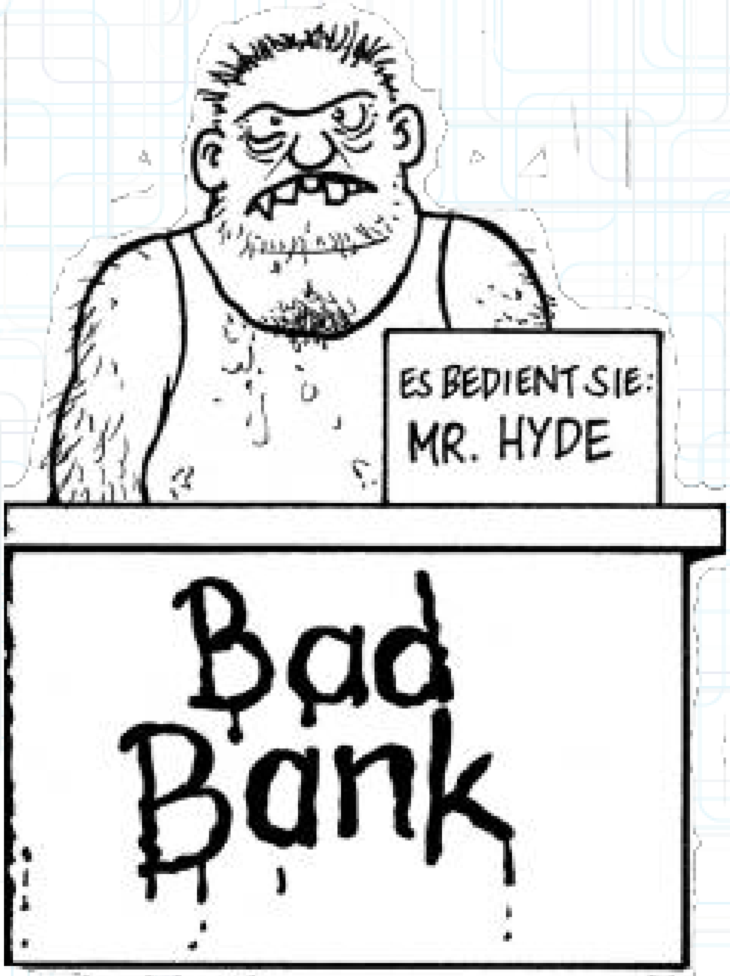
LiquiBase: XML

```
<changeSet id="4" dbms="oracle">
    <sqlFile path="5.sql"/>
</changeSet>

<changeSet id="5" context="test">
    <sqlFile path="5.sql"/>
</changeSet>

<changeSet id="6" failOnError="false">
    <sqlFile path="6.sql"/>
</changeSet>
```

LiquiBase DSL

```xml
<changeSet id="1" author="andrei">
    <createTable tableName="person">
        <column name="id" type="int" autoIncrement="true"/>
        <column name="firstname" type="varchar(50)"/>
        <column name="lastname" type="varchar(50)"/>
    </createTable>
</changeSet>
```

DEMO



https://github.com/asolntsev/liquibase-ee

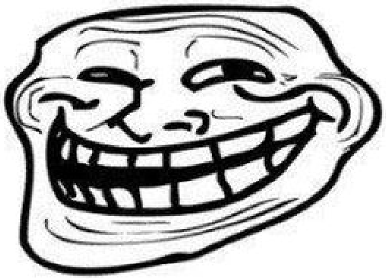
LiquiBase DSL

```
<changeSet id="1" author="andrei">
    <createTable tableName="person">
        <column name="id" type="int" autoIncrement="true"/>
        <column name="firstname" type="varchar(50)"/>
        <columr
    </createTat
</changeSet>
```

- Supports all Dbs
- IDE autocompletion
- Rollback out-of-the-box
- Built-in refactorings

LiquiBase refactorings

```
<changeSet>
   <renameColumn
      tableName="order_comment"
      oldColumnName="author_id"
      newColumnName="employee_id"
   />
</changeSet>
```



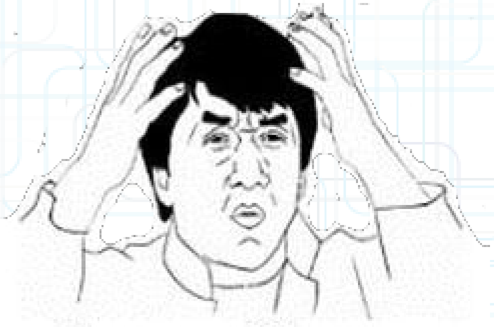
LiquiBase refactorings

```
<modifyDataType
    tableName="customer"
    columnName="middle_name"
    newDataType="VARCHAR2(100)"
/>
```

Data loading

```
<changeSet runOnChange="true">
    <loadData tableName="CLIENT" file="clt.csv">
        <column header="id"
                  name="ENTITYID"/>
        <column header="DESCRIPTION"
                  name="DESCRIPTION"/>
    </loadData>
</changeSet>
```

Why it's needed?



Oppa Agile style

- Local development
    - Oracle XE (dev servers must die!)
- Easy run
    - ant start
- In-memory DB for tests/demo
    - ant test

3 DBs

1. Oracle - in production
2. Oracle XE - in development
3. H2 in-memory

- for demo

- for tests

- for designer

DB configuration in GIT

conf/

- dev.properties
  - db.url=jdbc:oracle:thin:@server1:1521:devdb
- local.properties
  - db.url=jdbc:oracle:thin:@**127.0.0.1:1521:xe**
- inmemory.properties
  - db.url=**jdbc:h2:out/liquibase-ee**

Run

- ant
- ant -Denv=dev
- ant -Denv=live
- ant start - runs demo

Opposition

SQL men: it works only for Java

CVS men: we already have scripts

Admins: we wanna see SQL!

SQL preview

ant show-sql

LiquiBase

generates,

but

does not run SQL

Admins are satisfied

Procedures

```
<changeSet runOnChange="true" dbms="oracle">
   <sqlFile
        path="procedures/log_message.sql"
        splitStatements="false"/>

   <rollback>
        DROP procedure log_message
   </rollback>
</changeSet>
```

Simple DB refactorings

My favorite - having

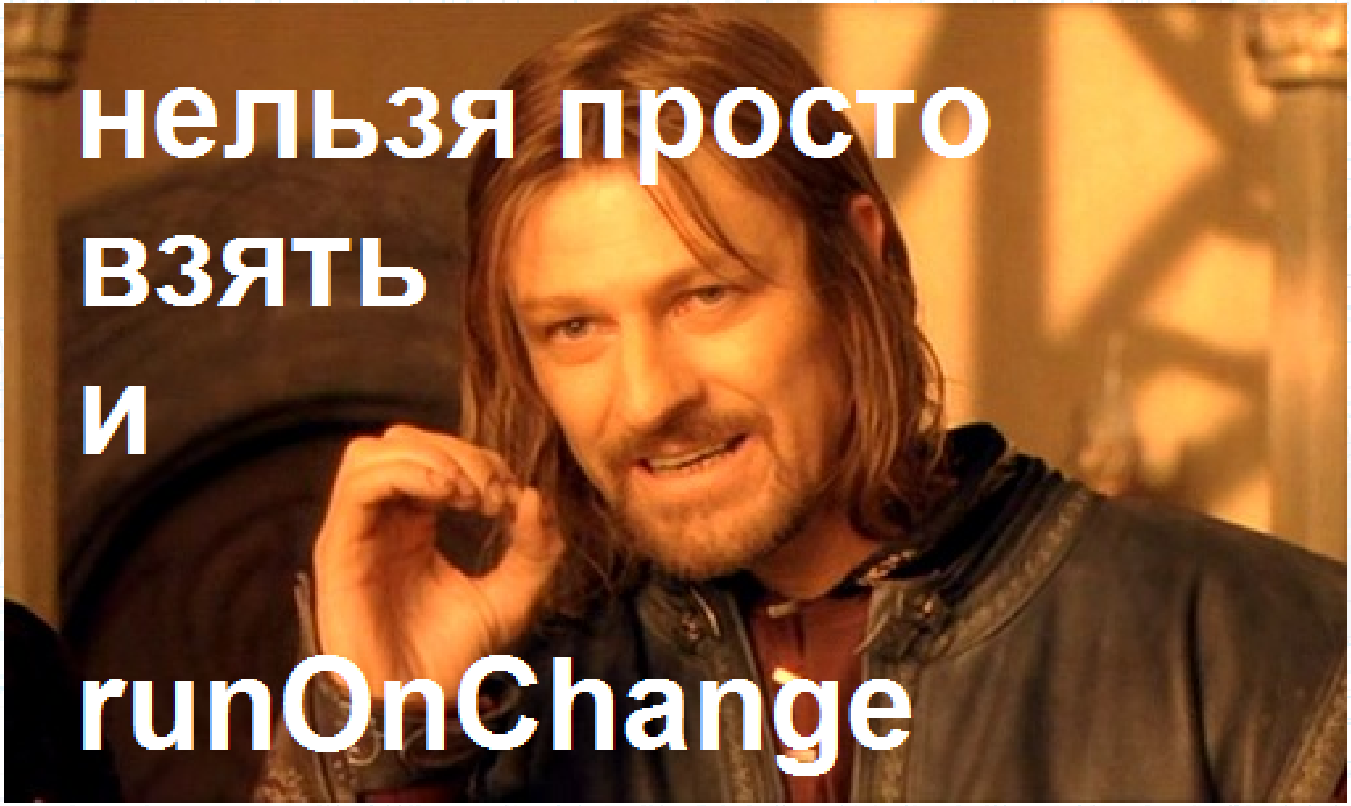
“CREATE OR REPLACE”:

- Procedures
- Functions
- Pl/Sql
- VIEWs
- Triggers

```
<changeSet runOnChange="true">
    <sqlFile path="another.sql"/>
</changeSet>
```

Nontrivial refactorings

Those without
CREATE OR REPLACE



Materialized view

```
<changeSet id="002">
<sql>
    CREATE MATERIALIZED VIEW currency_mv ...
</sql>
</changeSet>
```

Materialized view

```
<changeSet id="001" failOnError="false">
<sql>
    DROP MATERIALIZED VIEW currency_mv;
</sql>
</changeSet>
```

```
<changeSet id="002">
<sql>
    CREATE MATERIALIZED VIEW currency_mv ...
</sql>
</changeSet>
```

We are happy

- It's easy to develop DB

We are happy

- It's easy to develop DB
- Jenkins tests DB scripts

We are happy

- It's easy to develop DB
- Jenkins tests DB scripts
- Designer edits HTML

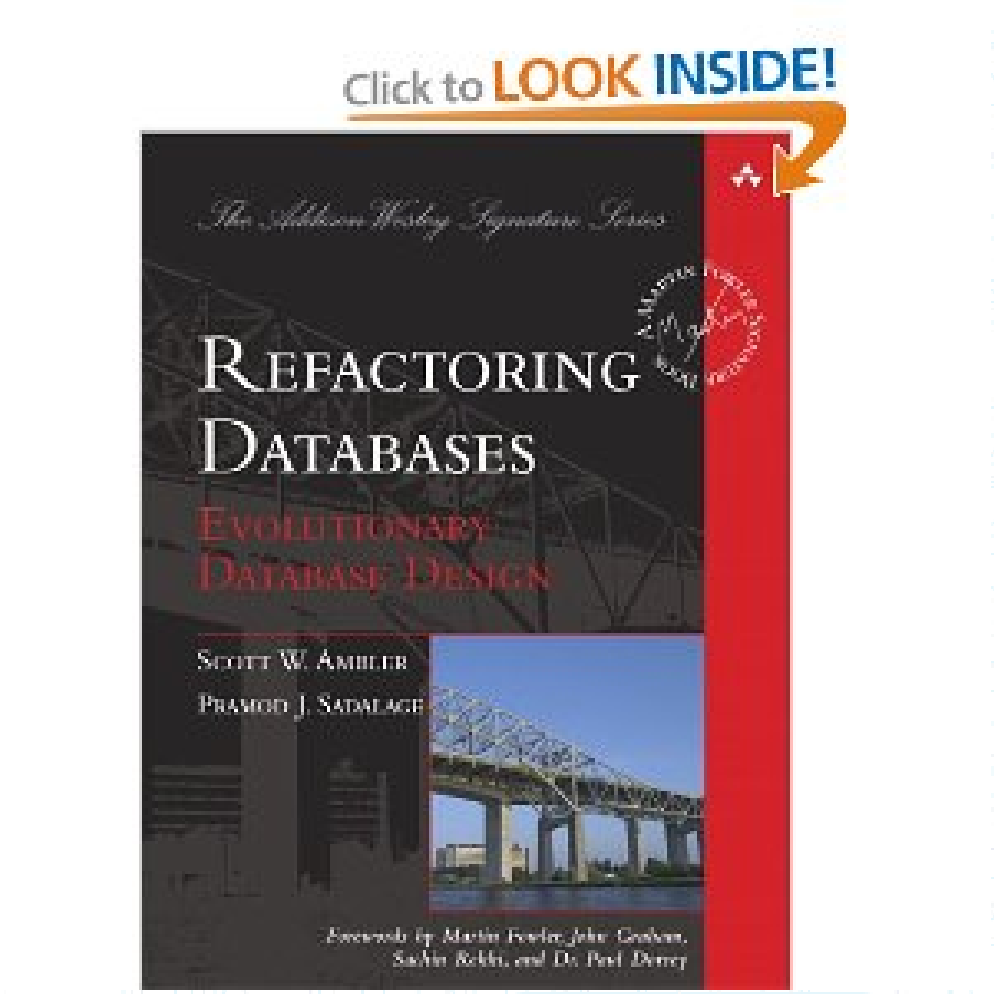
We are happy

- It's easy to develop DB
- Jenkins tests DB scripts
- Designer edits HTML
- Admin sleeps at night

We are happy

- It's easy to develop DB
- Jenkins tests DB scripts
- Designer edits HTML
- Admin sleeps at night
- UI-tests run on localhost

To read


Click to LOOK INSIDE!
The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Refactoring Databases
Evolutionary Database Design
Scott W. Ambler
Pramod J. Sadalage
Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dorsey

@asolntsev

commit & push

Andrei Solntsev
andrei.solntsev@gmail.com

//codeborne
well-crafted software

Selenide
CONCISE UI TESTS IN JAVA

Behind the scenes

changelog.xml

releases

```
<databaseChangeLog>
    <include file="changelog_001.xml"/>
    <include file="changelog_002.xml"/>
    <include file="changelog_003.xml"/>
</databaseChangeLog>
```

Raw SQL

```
<changeSet id="201308181029">
    <sql>
        CREATE TABLE payment (
            id NUMBER NOT NULL,
            description VARCHAR2(100 CHAR),
            url VARCHAR2(500 CHAR),
            upd_dttm TIMESTAMP)
        </sql>
        <rollback>
        DROP TABLE payment
        </rollback>
</changeSet>
```

Constraints

```
<changeSet>
    <addColumn tableName="payment">
        <column name="customer_id" type="NUMBER"/>
    </addColumn>

    <addForeignKeyConstraint
        baseTableName="payment"
        baseColumnNames="customer_id"
        constraintName="payment_customer_fk"
        referencedTableName="customer"
        referencedColumnNames="id"/>
</changeSet>
```

Some DB-specific (e.g. Oracle)

```
<changeSet dbms="oracle">

    <createSequence
        sequenceName="order_file_id_seq"/>

</changeSet>
```

Materialized view

```
<changeSet dbms="oracle">
<sql>
    CREATE MATERIALIZED VIEW currency_mv
    REFRESH COMPLETE
    as select * from (select * from
        currency_xa_pos2@${db.dblink.core})
</sql>
<rollback>
    DROP MATERIALIZED VIEW currency_mv;
    </rollback>
    </changeSet>
```

Refresh Mat View

```
<changeSet dbms="oracle">
    <sqlFile path="mviews/refresh_mv.sql"/>
    <rollback><sql>
        BEGIN
            dbms_refresh.destroy('CUR_GRP');
        END;
    </sql></rollback>
</changeSet>
```

Refresh Mat View

```
<changeSet dbms="oracle">
    <sqlFile path="mviews/refresh_mv.sql"/>
    <rollback><sql>
        BEGIN
            dbms_refresh.destroy('CUR_GRP');
        END;
    </sql></rollback>
</changeSet>
```

LiquiNinja

How to re-run changesets:

```
<changeSet id="115">
    <sql>
        update databasechangelog
        setfilename = 'db/static_data_001.xml',
                md5sum = null
        where id = '201309261214'
    </sql>
</changeSet>
```

@asolntsev

commit & push

Andrei Solntsev
andrei.solntsev@gmail.com

codeborne
well-crafted software

Selenide
CONCISE UI TESTS IN JAVA