Liquibase

Nicholas Wertzberger

What are we talking about?

- SQL
- Managing SQL
- Liquibase
- Managing liquibase

SQL

```
create table country (
  id varchar(55) not null
);
```

SQL

```
alter table state
   add country varchar(55)
   default ‘USA’;
```

SQL

```
alter table state
 add constraint state_fk1
 foreign key (country)
   references country(id);
```

Managing with SQL

- We want
  - Straightforward format for changes and rollbacks
  - Rollback testing

Constraints with Data

- Non-reversible changes
- Non-repeatable changes

Ideas

- SQL should be treated like code
- Have consistent way to
  - manage changes
  - roll back
  - recreate

Liquibase

```
<databaseChangeLog>
  <include file="ddb.changelog-1.0.0.xml"/>
  <include file="db.changelog-1.0.1.sql"/>
</databaseChangeLog>
```

Liquibase

```
<createTable tableName="state">
  <column name="id"
          type="varchar(55)">
    <constraints nullable="false"/>
  </column>
</createTable>
```

Liquibase

```
<addPrimaryKey
   tableName="person"
   columnNames="id"
   constraintName="person_pk"/>
```

Liquibase

```
<createIndex tableName="person"
             indexName="person_x1">
    <column name="firstname"></column>
    <column name="lastname"></column>
</createIndex>
```

Liquibase

```
<sql>
alter table lob_table
  add (sample clob not null)
lob (sample) store as securefile schema_lob_table_ls1(
  tablespace LOB_TS
  disable storage in row
  index schema_lob_table_li1(tablespace LOB_TS)
)
</sql>
```

Changesets

- Labels
- Contexts
- author:id

```
<changeSet
  id="2"
  author="nwertzberger"
  context="DEV"
  labels="STGCIN-122">
</changeSet>
```

Liquibase

- DEMO
  - Apply
  - Tag
  - Roll back
  - What if rollback fails?
  - Changelog

Liquibase

- Additive format
- Same artifact for all environments
- Common rollback format
- Extensible markup

What else is still wrong?

- Rollback testing
- Refactoring

Managing Liquibase

- Create a preview database
  - Same structure as development
  - Backup before deploy
  - Restore on failure
  - Integrate into change pipeline
  - Run deploys with **"updateTestingRollback"**

Managing Liquibase

- Make snapshot and recovery easy
  - Rollbacks will fail
  - Two options: recovery, liquibase surgery

Managing Liquibase

- Don't do "irreversible" changes
  - Drops can be rename's
    - TABLE_NAME_DROP
    - COLUMN_NAME_DROP
  - Hide drops in "IRREVERSIBLE" context.
    - Always specify a context on updates

Refactoring Liquibase

- If you change these on an already-ran change, **you will break liquibase**
  - filename
  - author
  - id
  - anything about the change

Refactoring Liquibase

- clearCheckSums
  - Clears md5 sums of ran changes
  - Lets you change what that changeset does
- changeLogSync
  - Adds changesets to changelog without running it

Getting Started

- generateChangeLog
  - Create changelog from existing database

Takeaways

- Liquibase
  - probably better than what you're doing today
  - Extensible markup
  - Still hard to refactor