VISCOSITY
PROFESSIONAL SERVICES

ORAPUB
POWERED BY VISCOSITY

Liquibase with Oracle
Automate your schema management

NLOUG Database Cloud & Developer Day
Nov 13, 2025

NL.OUG
NL ORACLE USER GROUP

Gary Gordhamer

Oracle ACE Director
Managing Principal Consultant

@ggordham.bsky.social
linkedin.com/in/ggordhamer/
gary.gordhamer@viscosityna.com

viscosityna.com
@ViscosityNa

Gary Gordhamer

Oracle ACE Director
Managing Principal Consultant

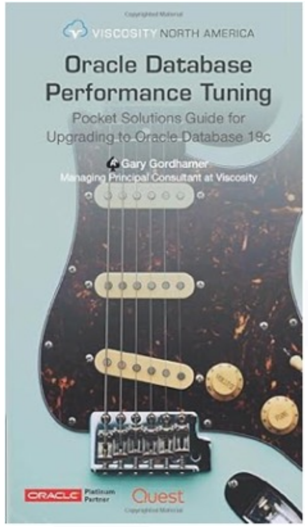
@ggordham.bsky.social
linkedin.com/in/ggordhamer/
gary.gordhamer@viscosityna.com

30+ years with Oracle technology

- Database 6.x – 23c
- Oracle Financials 7, E-Business Suite R11, R12.1, R12.2
- IDM integrations and Security hardening

Worked in many industries including

- manufacturing
- financial
- marketing
- healthcare
- utilities
- Government



VISCOSITY NORTH AMERICA
Oracle Database Performance Tuning
Pocket Solutions Guide for Upgrading to Oracle Database 19c
Gary Gordhamer
Managing Principal Consultant at Viscosity
ORACLE Platinum Partner
Quest

Wrote the book:
"Oracle Database Performance Tuning:
Pocket Solution Guide Series For Upgrading Oracle Databases"

Active member of the Oracle user community presenting frequently!

Oracle ACE Director

viscosityna.com

@ViscosityNa

Viscosity Pillars and Delivery Models

DATA

Oracle & SQL Server PostGres

Performance Tuning

Data Replication

Data Warehousing Analytics

Data Integration

ERP Blue Prints

Database Upgrades

APPS

APEX

EBS

Web/Mobile Apps

.Net and C#

E-Business Suite

SAAS/PAAS

CLOUD

Azure Gold Partner

Cloud Migrations

Engineered Systems

Oracle Cloud Partner

Google Partner

AWS Partner Hybrid Cloud

Workshops

Assessments

Proof of Concepts

Training

Turnkey Projects

Managed Services

Viscosity’s Oracle ACEs
The Oracle ACE Program

The Oracle ACE Program recognizes and rewards individuals for their contributions to the Oracle community.

Charles Kim
CEO | Co-Founder

@racdba

ACE Director

Rich Niemiec
Chief Innovation Officer

@richniemiec

ACE Director

Craig Shallahamer
Applied AI Scientist

@orapub

ACE Director

Sean Scott
Principal Consultant

oraclesean.com

ACE Director

Gary Gordhamer
Principal Consultant

ggordham.bsky.social

ACE Director

Marco Pereira
Software Architect

@markuspg

ACE Associate

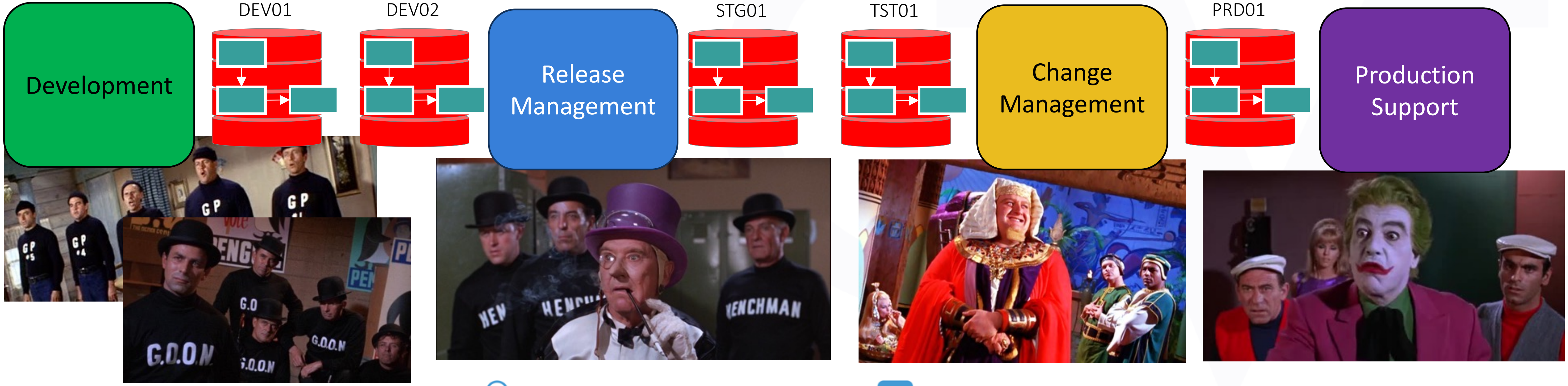
viscosityna.com

@ViscosityNa

Schema Management

Riddle me this:

- How do you keep track of changes?
- How do you keep multiple environments in sync?
- How do you KNOW things match?



Development

DEV01

DEV02

Release Management

STG01

TST01

Change Management

PRD01

Production Support

Current tools?

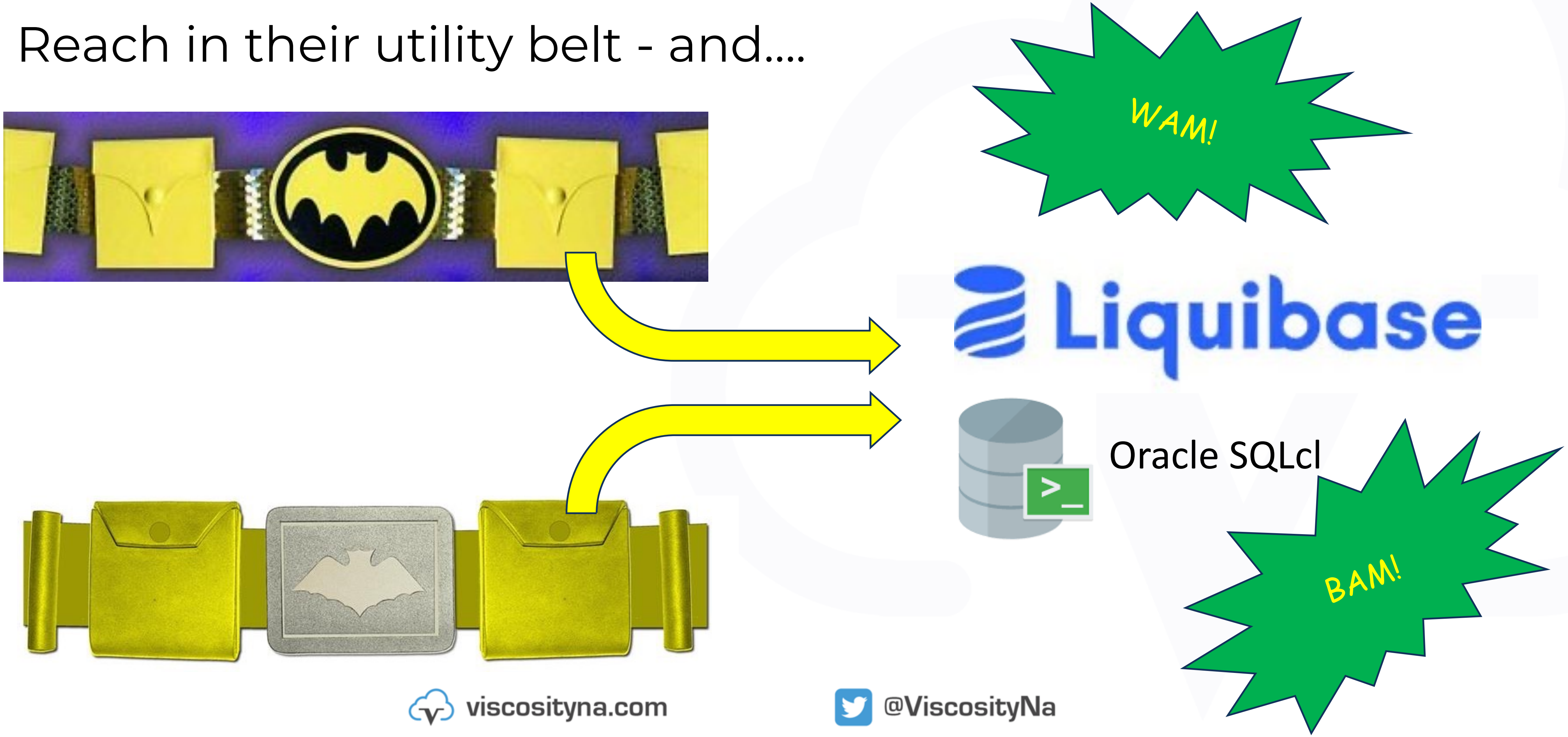
- SQL Developer
- Quest Toad
- Home Grown
- Developers?

Does your tooling keep your schemas in-sync?

TRAPPED?

What will our hero do?

Reach in their utility belt - and....



Liquibase

Oracle SQLcl

What is Liquibase?

- Opensource tool started in 2006 for tracking, managing and applying database changes
- Opensource and ”Pro” licensed editions
- Oracle SQLcl includes the Opensource edition
- Updated frequently adding features
- Demo is using:
  - SQLcl 24.3.2 / Nov 2024
  - Liquibase 4.25

You manage at the SCHEMA level.

Liquibase Commands

- `generate-schema` - Writes Change Log XML to copy the current state of the database to files.
- `generate-db-object` - Writes Change Log XML to copy the current state of the database object to a file, used to update objects.
- `update` - Deploy any changes in the changelog file that have not been deployed.
- `update-sql` - Generate the SQL identified in the changelog for review before running the update command.
- `tag` - Mark the current database state with the specified tag to use for roll back.
- `rollback` - Rollback changes made to the database based on the specified tag.

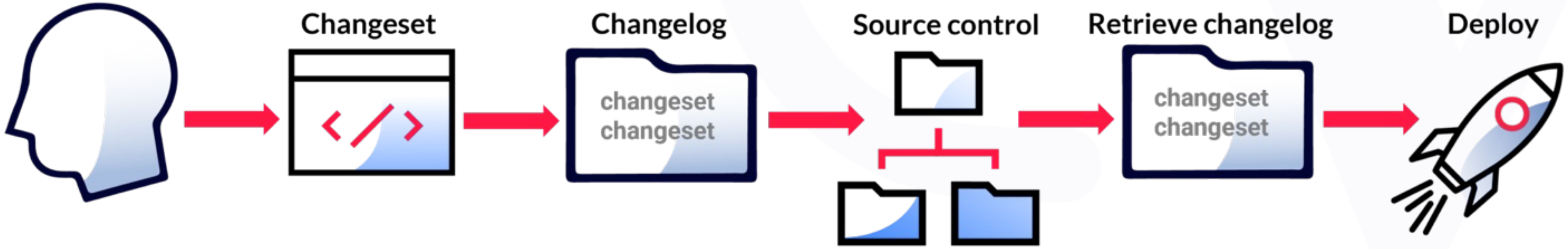
Changelog

Unit of Liquibase change

- Can contain one or more changes
- Can be in one file or include multiple files
- Default format is XML, can also use SQL, YAML, JSON

Changeset – uniquely tagged atomic change

- ID = unique id of this change
- AUTHOR = who
- Settings: failOnError, runOnChange, runAlways, runOrder, etc.



ChangeLog example – abbreviated

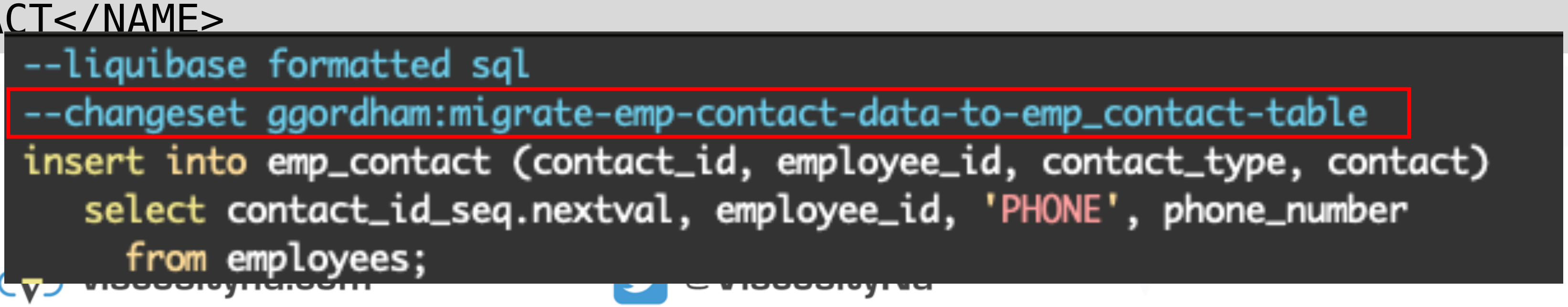
controller.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog>
  <include file="emp_contact_table.xml"/>
  <include file="contact_id_seq_sequence.xml"/>
  <include file="cr_contact_data.sql"/>
  <include file="upd_emp_contact_proc_procedure.xml"/>
</databaseChangeLog>
```

emp_contact_table.xml

```
<changeSet id="6705526b34ad9a8dc85e41c56583f2da74370d7c"
author="(HR_DEV)-Generated" failOnError="false” runOnChange="true”
runAlways="false"  >
<n0:createSxmlObject objectName="EMP_CONTACT" objectType="TABLE”
ownerName="HR_DEV"  replaceIfExists="true" > <n0:source><![CDATA[
  <TABLE xmlns="http://xmlns.oracle.com/ku" version="1.0">
   <SCHEMA>HR_DEV</SCHEMA>
   <NAME>EMP_CONTACT</NAME>
```

cr_contact_data.sql



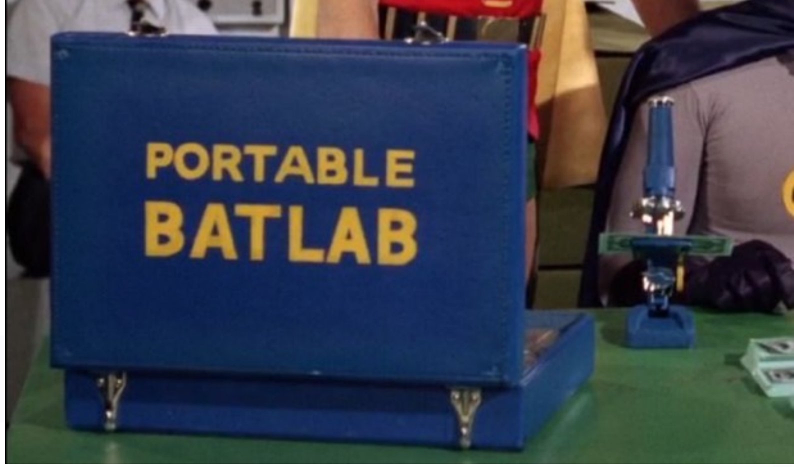
```
--liquibase formatted sql
--changeset ggordham:migrate-emp-contact-data-to-emp_contact-table
insert into emp_contact (contact_id, employee_id, contact_type, contact)
   select contact_id_seq.nextval, employee_id, 'PHONE', phone_number
     from employees;
```

Changes stored in DB tables

- DATABASECHANGELOG
  - Track which database changes have been run
  - Rolling back a change removes it from this log
- DATABASECHAGNELOGLOCK
  - Prevents simultaneous runs of Liquibase
- DATABASECHANGELOG_ACTIONS
  - Tracks object states and SQL statements during deployment

Note: these tables are at each SCHEMA level!

Methodology



- CI/CD (*in-place*) – DEMO 1
  - Have a core set of scripts that can be run at anytime
  - Will re-build everything to a point in time
  - Can have objects and seed data
  - Major issue – merging changes
- Waterfall (*by version*) – DEMO 2
  - Have different scripts for each change / release
  - Would never (or rarely) re-build everything
  - Goal is to maintain drift / change
  - Major issue – tracking all changes properly

Methodology – part 2

- Can all changes go through Liquibase?
  - Development writes changes in LB code
  - Can execute or rollback LB code in development
  - Must execute or rollback LB code in TEST and PROD
  - Tags code to keep it separate from other developers work
  - Issue – LB log overlap of code changes
- What if development is "wide open"?
  - Changes are captured to LB code when ready
  - LB code is promoted to test, and test is 100% LB maintained
  - Issue – when DEV is refreshed you lose work in progress

Demo 1 Structure – CI /CD

```
hr
├── comment
├── index
├── procedure
├── ref_constraint
├── sequence
├── table
├── trigger
└── view
```

Example structure

- Schema name (HR)
- Base – original capture / baseline
- Check all files into source code control (GIT)
- Update development, and “rebuild” the scripts using Liquibase
- Source code control used to identify changes

NOTE: special use cases for rollback of PL/SQL code, will discuss in demo 2.

Directory Structure - Waterfall

```
hr
├── comment
├── index
├── procedure
├── ref_constraint
├── sequence
├── table
├── trigger
├── view
└── waterfall
        ├── v1.1
        ├── v1.2
        ├── v1.3
        └── workarea
```

Example structure

- Schema name (HR)
- Base – original capture / baseline

- Each released version
- Work area to capture new scripts

Demonstration steps

- Baseline from PRODUCTION
  - Generate XML files from production
  - Mark production as “changes have been run”
- Compare DEVELOPMENT to PRODUCTION
- Generate changes from DEVELOPMENT
- Prepare TEST
  - Refresh TEST from PRODUCTION
  - Apply DEVELOPMENT changes
  - Rollback DEVELOPMENT changes
- Prepare for production

Want to try this out?

Perrrfect DEMO

You can download all the scripts from GitHub

https://github.com/ggordham/ora-presentations/tree/main/liquibase-NLOUG

All scripts are "as is" and meant for teaching purposes only.

DO NOT run these in any production system or system relied upon by your development teams.

Demo environment

Three levels (PDBs)

- Production – hrprd
- Test – hrtst
- Development – hrdev

Schema: HR

Make changes / build out in Development

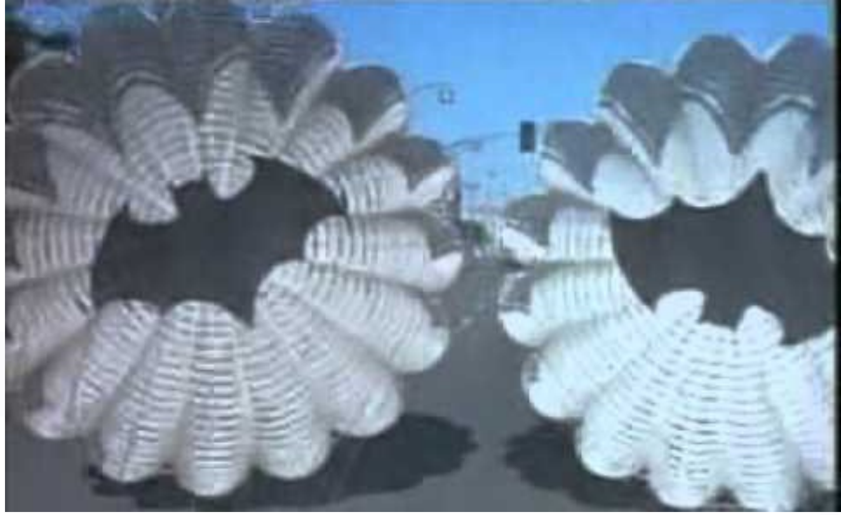
Test releases in Test

Production – source for refresh back to DEV / TST

High level process

1. Baseline / create original schema
2. Code changes in DEV
3. Capture changes to XML or SQL
4. Collect changes to a changelog / version
5. Check version into code repository
6. Deploy version
7. Rollback version if needed

Notes on issues



What if Liquibase gets out of sync?

Options

- Refresh the environment and re-apply “working” scripts
- Manually fix the issue
  - Resolve issue with DB objects
  - Resolve records (remove?) from DATABASECHANGELOG table

Release Management

We need to refresh test and get all the changes loaded

No problem!

- Refresh the DB from production (DP / RMAN)
- Grab changes from source control (GIT)
- Run: `lb update -changelog-file controller.xml`

Need to load custom data or run data cleansing scripts?

- SQLcl – data loading built in (CSV)
- SQLcl – runs all SQL*Plus scripts

Change Management

What's your release plan?

How was this tested?

What's your backout plan?

SOCK!

No Problem

- Running the SQLcl scripts from development and in GIT
- The same scripts were executed in TEST
- If needed, run:

```
lb rollback -changelog-file controller.xml -tag v1.5
```

Summary - Liquibase

- Automate all your schema changes
  - Track changes
  - Build deployment scripts
  - Customize as needed (pre-scripts / post-scripts / rollback)
- Built into SQLcl
  - Free to download / use
  - Pre-installed with SQL-Developer
  - Standalone install – also in every ORACLE_HOME
  - Opensource - frequently updated

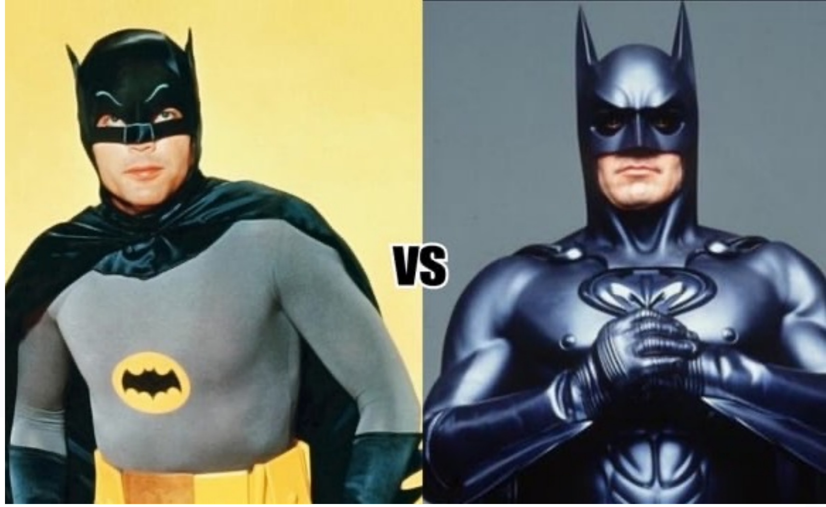
You don't have to be a billionaire to be super!

SQLcl
Liquibase vs. Projects

Projects is a new feature in version 24.3

"Application CI/CD" – `project` command

One-shot / easy button for APEX development



	Liquibase	Projects
Control	Full control	Minimal (configuration / custom scripts) Cannot control folder structure
Usage	Multiple methods additional commands: compare, test, diff, etc.	Simple - three main commands stage / release / deploy
Rollback	Automated or custom rollback available	Not available (move forward only)
Version Control	Recommended – can use any tool	Required – must use GIT
Code method	Can capture from DB, or manually author	Change database – then capture
Misc	Multiple database Note: SQLcl can capture APEX code also	APEX code capture built-in, Oracle only

Q&A – Additional information

Few additional SQLcl commands

Export all the data in a schema:

```
liquibase data -ovf -output-file data/hr-DATA.xml
```

Export specific table:

```
liquibase data -ovf -include-objects EMPLOYEES
    -output-file data/hr-employees-DATA.xml
```

Turn off DDL storage attributes when generating DDL:

```
SET DDL storage off
SET DDL segment_attributes off
SHOW DDL
```

Export an APEX application or ORDS module:

```
liquibase generate-apex-object -applicationid 101 -exporiginalids -split
liquibase generate-ords-module -module-name myModule
```

Dealing with identity columns

Data tables loaded from changesets will not update the identity column initial value.

Add a custom script to fix the identity columns

custom runOracleScript changeset

```
 <include file=”fix-identity-cols.sql"/>
```

```
/* fix-identity-cols.sql */
BEGIN
FOR i IN (SELECT table_name, column_name FROM user_tab_identity_cols) LOOP
EXECUTE IMMEDIATE 'ALTER TABLE '||i.table_name||' modify '||i.column_name
||' generated as identity start with limit value';
END LOOP;
END;
/
```

Thanks to Simon Pane for this tip!

Liquibase vs Oracle Liquibase

Oracle “extended” Liquibase to work with Sxml objects

- Sxml objects are the output of DBMS_METADATA
- They also support “updates” to objects (not just drop and re-create)

The required JAR files for this extension are included with SQLcl

- dbtools-liquibase.jar (sqlcl/lib/ext)
- dbtools-apex.jar (sqlcl/lib/ext)
- guava-with-lf.jar (sqlcl/lib)
- xmlparserv2_sans_jaxp_services.jar (sqlcl/lib)
- dbtools-common.jar (sqlcl/lib)

Update the liquibase.properties file, adding the following line:

```
change-exec-listener-class: liquibase.changelog.visitor.OracleActionChangeListener
```

Preconditions

Ability to check prior to execution of a changelog

  - changesetExecuted
  - tableExists, columnExists, viewExists, indexExists
  - sqlCheck

- More options available
  - onFail, onFailMessage, onError, onErrorMessage
  - HALT, WARN, CONTINUE, MARK_RAN

```
<preConditions onFail="HALT">
  <runningAs username="liquibase"/>
 </preConditions>
```

Contexts

Tags that control commands

- Set in changeset
  - --changeset ggordham:1.1 context:test
  - <changeset id=“1.1” author=“ggordham” context=“test”>
- Can have logic in the context
  - V1.0 or test
  - !test
  - !test and !v1.1
- Context set when LB run
  - --context-filter=“test”
  - --context-filter=“test,v1.1”

Windows

Default character set is not UTF-8

You may lose international characters in your code

Solution: set Java option

```
set JAVA_TOOL_OPTIONS=-Dfile.encoding=UTF-8
```

Can be set in a script to start SQLcl or in your system Environment Variables

https://rafal.hashnode.dev/configure-liquibase-standalone-oracle-sqlcl-for-usage-with-utf-8-encoding

References

- SQLcl download: https://www.oracle.com/database/sqldeveloper/technologies/sqlcl/download/
- Oracle SQLcl Users Guide: https://docs.oracle.com/en/database/oracle/sql-developer-command-line/24.3/sqcug/index.html
- SQLcl using Liquibase: https://docs.oracle.com/en/database/oracle/sql-developer-command-line/24.3/sqcug/using-liquibase.html
- Liquibase best practices: https://docs.liquibase.com/concepts/bestpractices.html
- Liquibase Changelog: https://docs.liquibase.com/concepts/changelogs/home.html
- Liquibase rollbacks: https://docs.liquibase.com/workflows/liquibase-community/using-rollback.html
- GitHub – DevOPs DB Tools Examples and Samples https://github.com/oracle/oracle-db-tools/tree/master/devops

Follow Us Online!

Facebook.com/ViscosityNa

LinkedIn.com/company/Viscosity-North-America

@ViscosityNA

Viscosity North America

Viscosity_NA