

Flyway



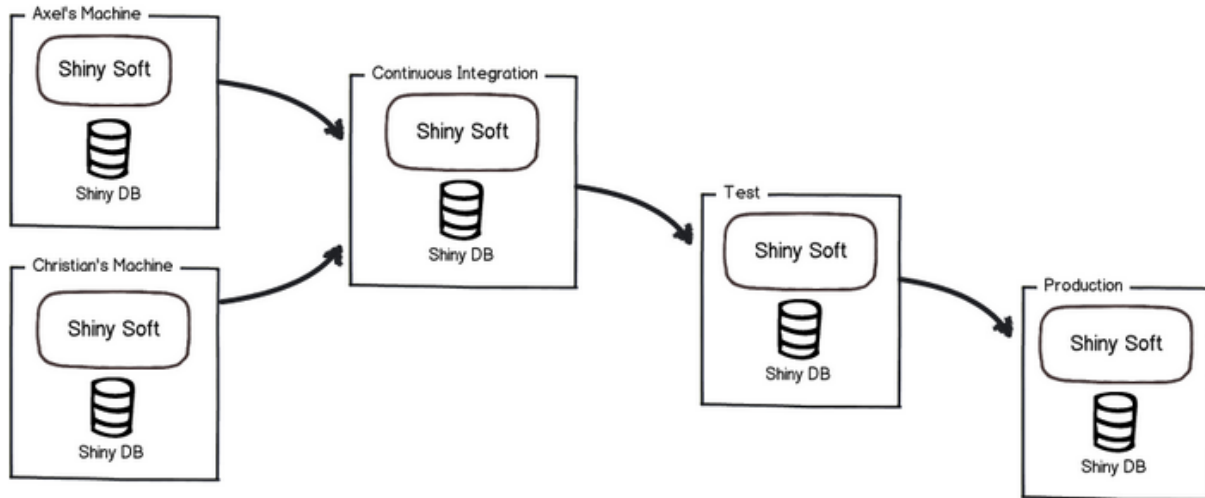
Kaunas JUG

The agile database migration framework for Java

Saulius Tvarijonas · saulius.tvarijonas@gmail.com

Why migration tool?

- Multiple environments (dev[1..*], staging, prod)
- Multiple developers



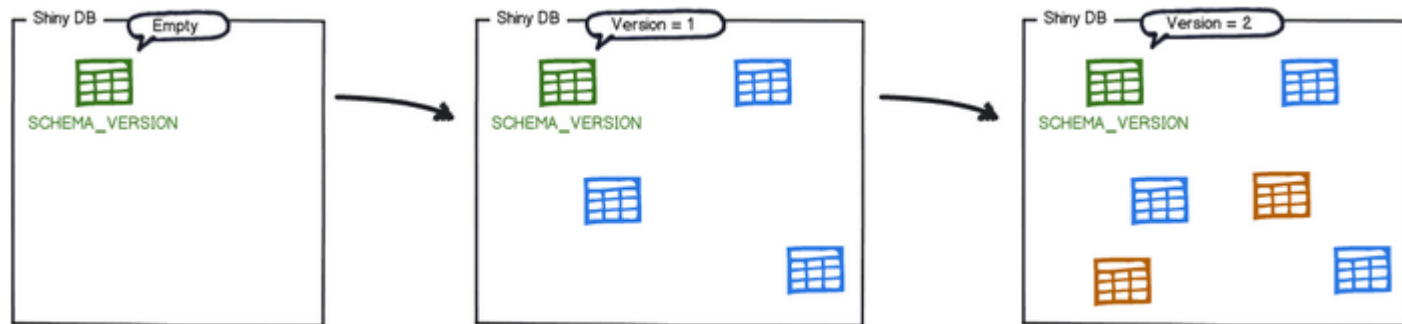
Questions?

- What is current state of database?
- Has script already applied or not?
- Has quick fix to production applied to test environment also?
- How do you set up a new database instance?

How it works?

- Metadata table ***SCHEMA_VERSION***
- Each change is recorded in metadata table
- Scans classpath for available migrations. SQL or Java.
- The migrations are then sorted based on their version number and applied in order

How it works?



How it works?

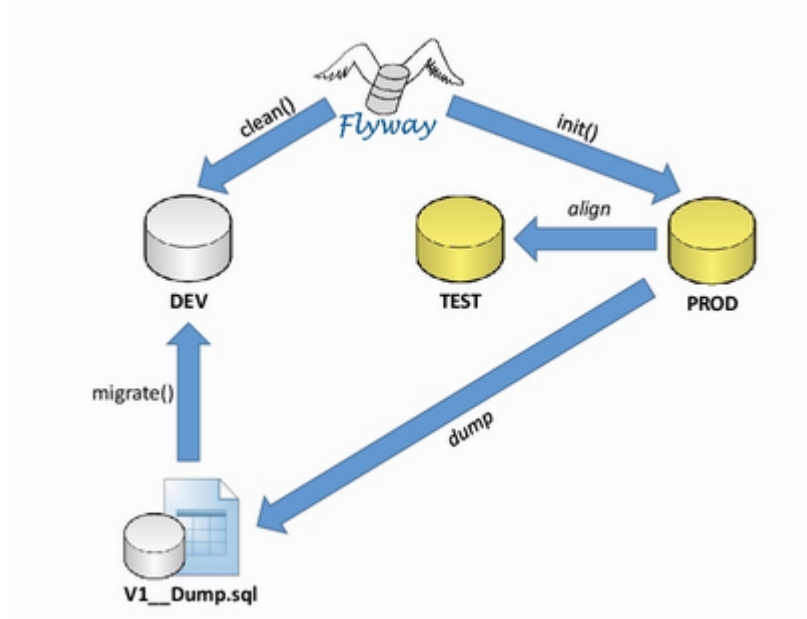
schema_version

version_rank	installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	1	1	Initial Setup	SQL	V1__Initial_Setup.sql	1996767037	axel	2010-05-04 22:23:00.0	546	true
2	2	2	First Changes	SQL	V2__First_Changes.sql	1279644856	axel	2010-05-06 09:18:00.0	127	true

Preparing an existing database

- Create an initial migration script that will recreate your current state and give it a low version number
- Use `flyway:init` to create the metadata table and set this script as the current version

Preparing existing database



Design Goals

- Cooperation between DEV and DBA
- Use full power of the DB
- Almost no lock-in
- Automatic so you can forget about it

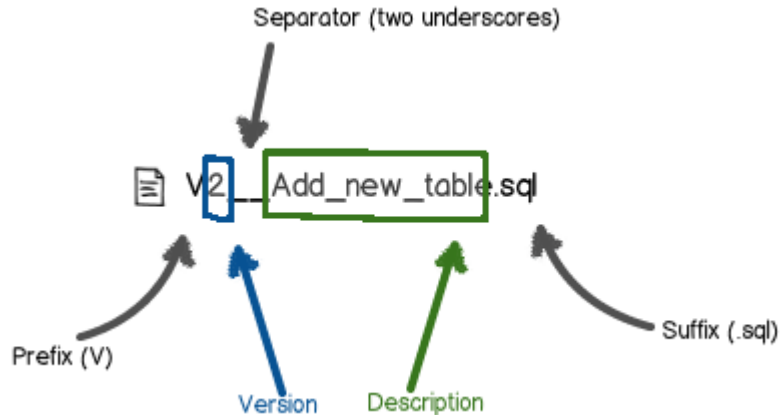
It just works

- Keep it simple
- Convention over configuration
- Zero required dependencies

Versions and naming

Examples of valid versions:

- 1
- 001
- 5.2
- 5_2 (5.2 at runtime)
- 1.2.3.4.5.6.7.8.9
- 205.68
- 20130115113556
- 2013.1.15.11.35.56
- 2013.01.15.11.35.56



Commands

INFO

INIT

MIGRATE

VALIDATE

CLEAN

REPAIR

Integrations

- Java API (also works on Android) for migrating the database on application startup
- Maven plugin, Gradle plugin, SBT plugin, Ant tasks or the Command-line client

Databases

Oracle

MariaDB

SQL Server

Google Cloud SQL

SQL Azure

PostgreSQL (inc. Amazon RDS and Heroku)

DB2

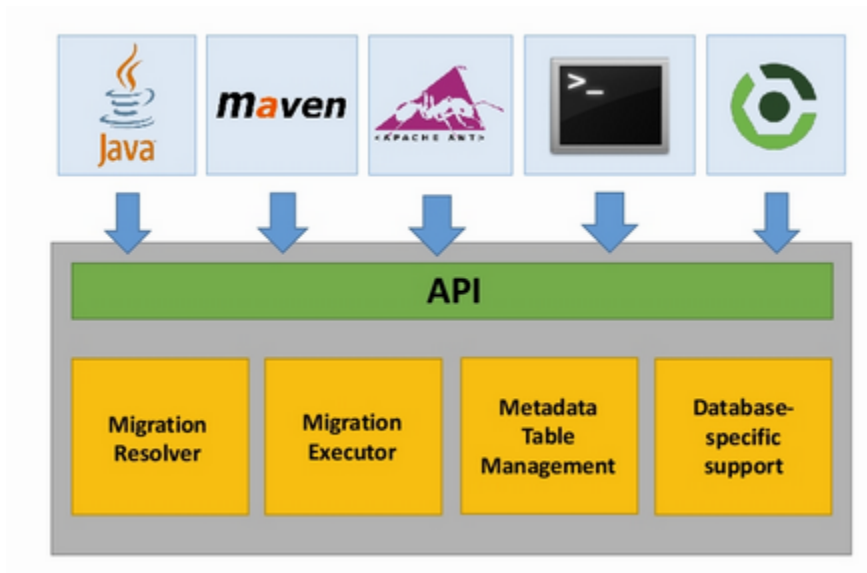
H2

MySQL (inc. Amazon RDS)

Hsql Derby

SQLite

Architecture



Java



```
Flyway flyway = new Flyway();  
flyway.setDataSource(url, user, password);  
flyway.migrate();
```

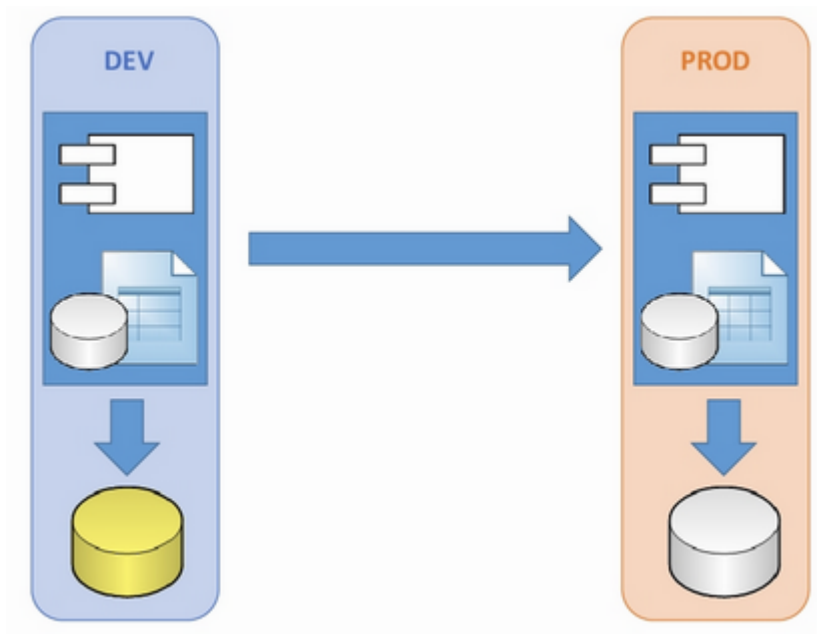

Maven



`mvn flyway:migrate`

```
<configuration>  
  <url>...</url>  
  <user>...</user>  
  <password>...</password>  
</configuration>
```

Apply to production



The Fear of DDL



DML \Leftrightarrow DDL



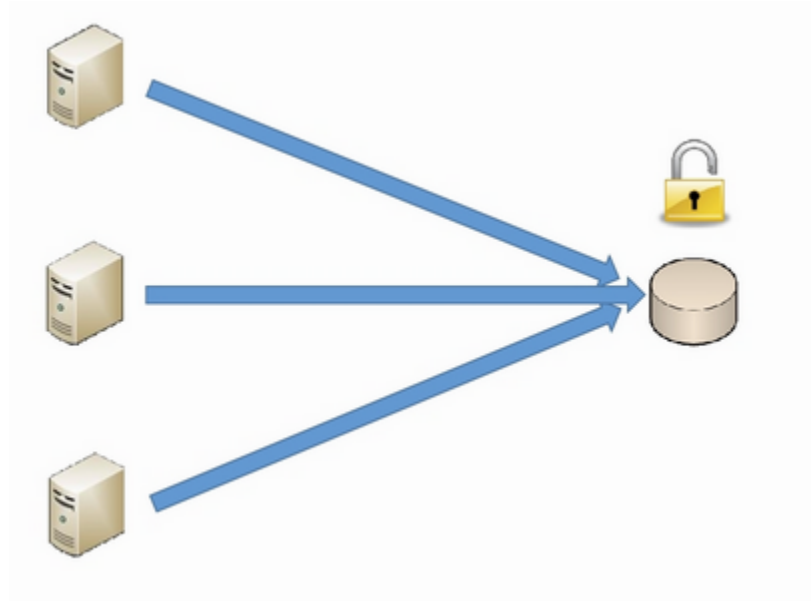
DML

VS



DDL

Cluster on start-up



When SQL isn't enough

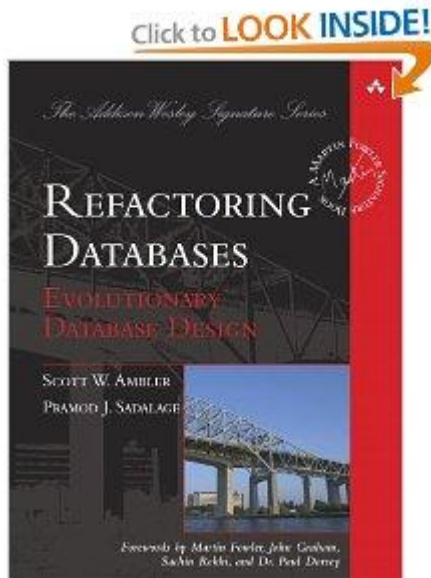
- Advanced text formatting
- BLOB



Java Migrations

```
public class V1_2__Another_user implements JdbcMigration {  
  
    public void migrate(Connection conn) throws Exception {  
  
        try (PreparedStatement stmt = conn.prepareStatement("...")) {  
            stmt.execute();  
        }  
    }  
}
```

Recommended



Java Forever

**Thank
You!**

