

XP Days Germany 2015

# Continuous Database Integration mit Flyway

Sandra Parsick

info@sandra-parsick.de

@SandraParsick

# Zur meiner Person

- Freiberufliche Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Softwerkskammer Dortmund



# Agenda

- Continuous Database Integration (CDBI)
- Flyway
- Flyway Demo
- Fallstricke

# Continuous Database Integration

- Definition
- Motivation
- Aufbau

# Definition

*„Continuous Database Integration (CDBI) is the process of rebuilding your database and test data any time a change is applied to a project's version control repository“*

(aus Continuous Integration by Paul M. Duvall, Steve Matyas und Andrew Glover)

# Motivation

- Alle Entwickler teilen sich eine Testdatenbank.
- Keiner weiß, welche Datenbankskripte auf welchen Datenbankinstanzen ausgeführt worden.
- Testdatenbank unterscheidet sich von der Produktionsdatenbank.
- Datenbankmigrationsskripte verteilen sich auf Emails, Release Notes, Ticketsysteme, etc.

# Aufbau

- Behandle den Datenbank-Code wie einen ganz normalen Source-Code
  - Alle Datenbank Artefakte (DDL, DML, Konfigurationen, Testdaten, Stored Procedures, Functions etc) gehören ins VCS.
  - Jede Änderung an den DB Artefakten wird getestet.
- Jeder Entwickler hat seine eigene Datenbank / Testdatenbanken ähneln den Produktionsdatenbanken.
  - Automatisiertes Aufsetzen der Datenbank.
- Änderungen an der Datenbank sind nachvollziehbar.
  - Historie der Änderungen

# Flyway

- Was ist Flyway?
- Wie funktioniert Flyway?
- Wie werden Migrationsskripte für Flyway geschrieben?
- Was kann Flyway nicht?
- Wie kann Flyway benutzt werden?

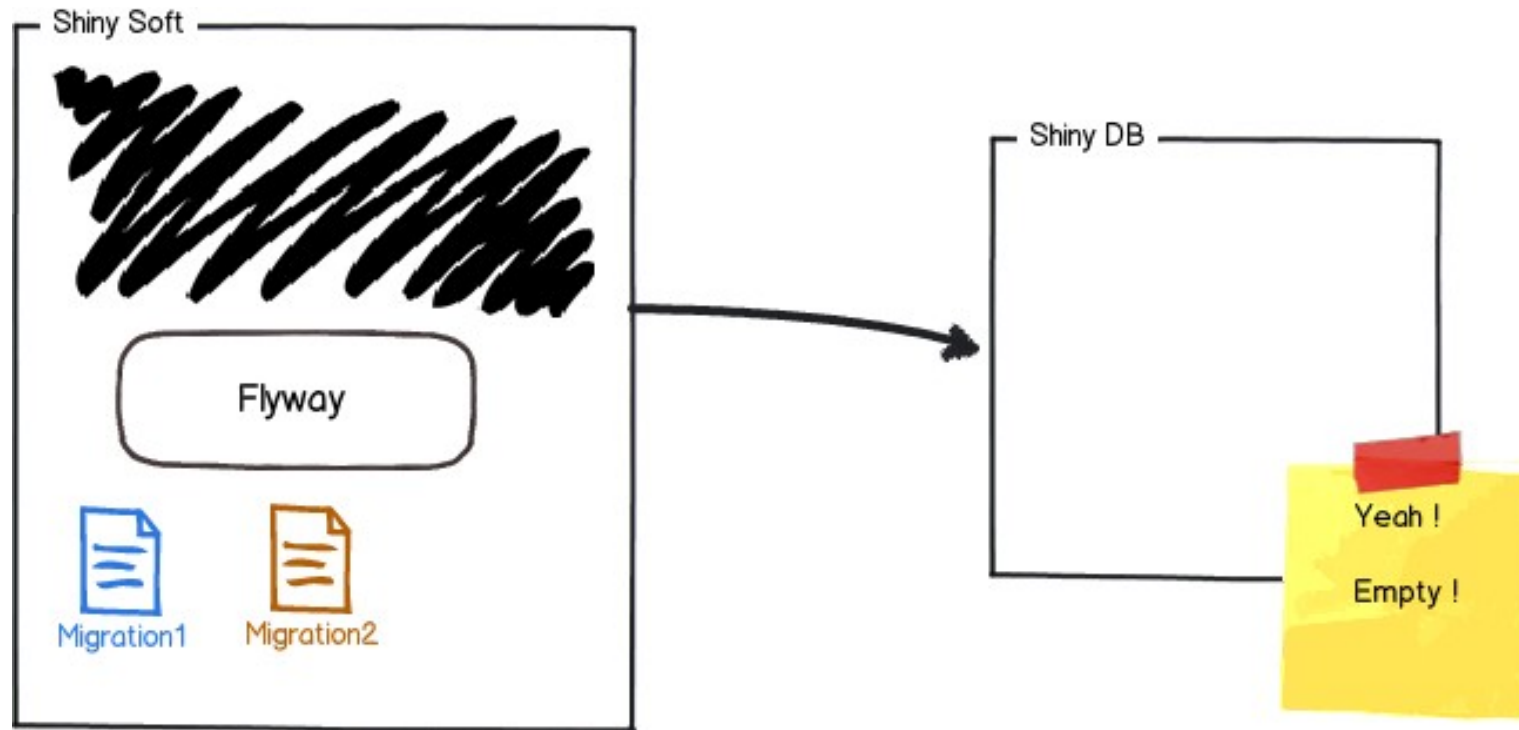


# Was ist Flyway?



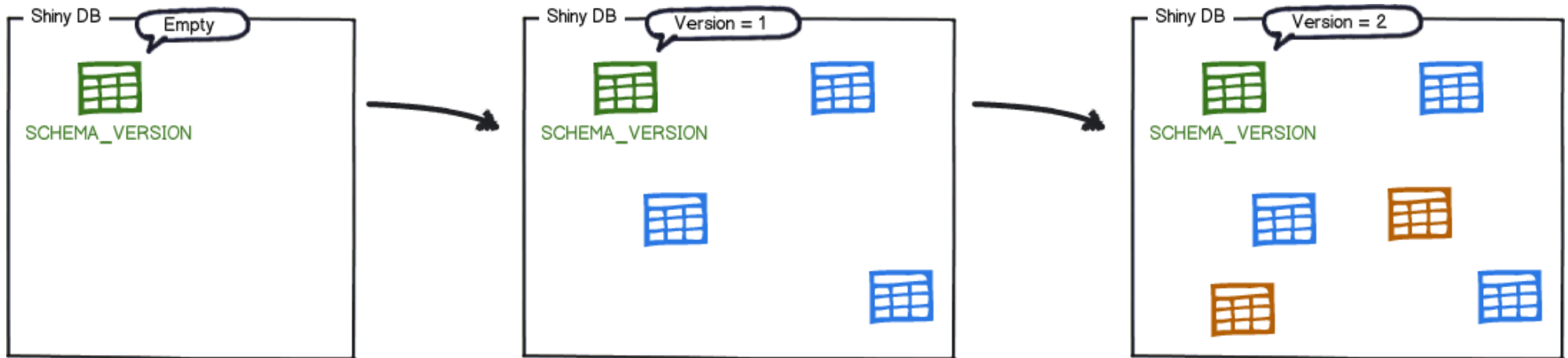
- Migration Framework für Relationale Datenbanken basierend auf Java
- Erstellt eine Datenbank „from scratch“
- Verwaltet den Stand der Datenbank
- Zwei Migrationsmodi:
  - SQL Migration
  - Java Migration
- Aktuelle Version: 3.2.1
- Homepage: <http://flywaydb.org/>
- Twitter: @flywaydb

# Wie funktioniert Flyway?



# Wie funktioniert Flyway?

migrate



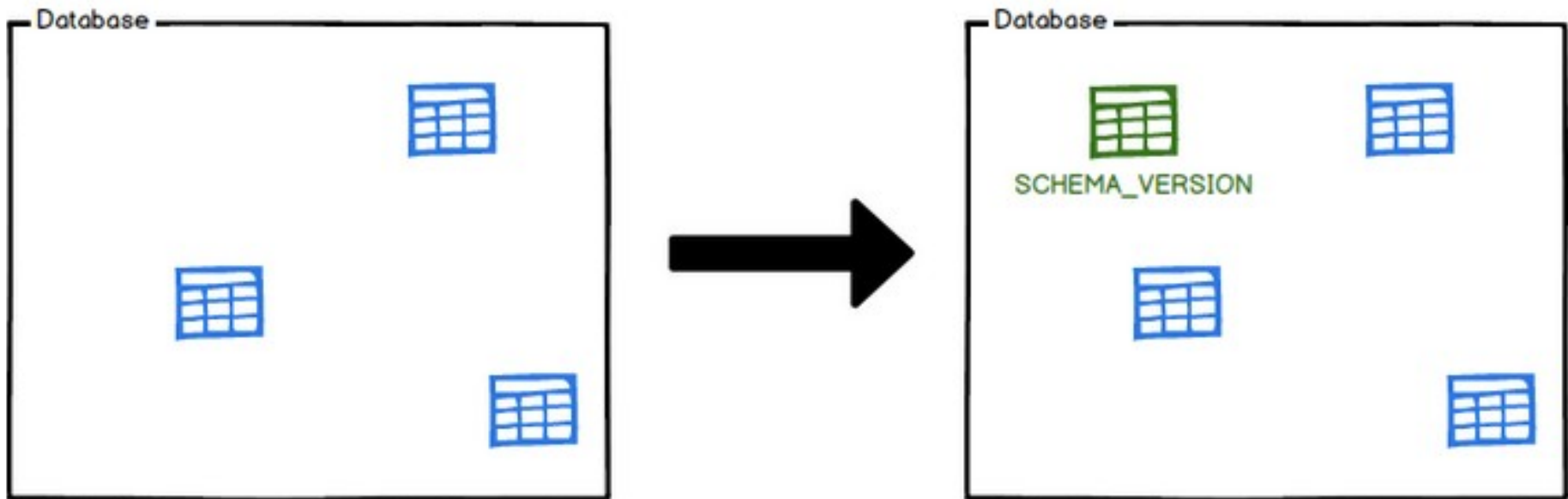
Reference: flywaydb.org

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

# Wie funktioniert Flyway?

baseline

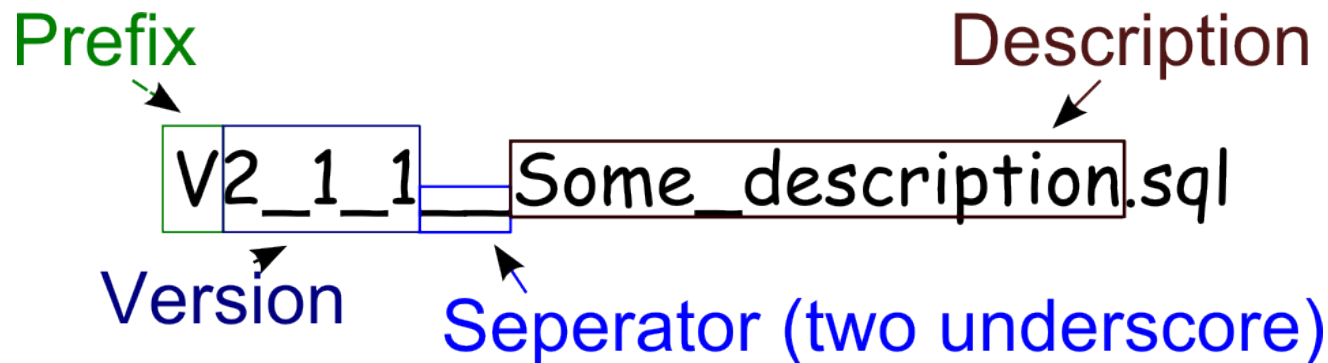


# Migrationskripte

- Zwei Möglichkeiten
  - SQL
  - Java

# SQL Migration

- **Typische Anwendungsfälle**
  - DDL Änderungen (CREATE/ALTER/DROP für TABLES, VIEWS, TRIGGERS, SEQUENCES, ...)
  - Einfache Datenänderungen
- **Benennung der Skripte**



# SQL Migration

- **Syntax**

- Statement kann über mehrere Zeile gehen
- Platzhaltersupport
- Kommentare: Single (–) oder Multi-Line (/\* \*/)
- Datenbank-spezifische SQL Syntax

- **Beispiel**

```
1  /* Create a table for person */
2
3  Create table person (
4      first_name varchar(128),
5      last_name varchar(128)
6  );
```

# Unterstützte Datenbanken



**Oracle**

10g and later (all editions, including XE)



**SQL Server**

2008 and later



**SQL Azure**

latest



**MySQL**

5.1 and later (incl. Amazon RDS)



**MariaDB**

10.0 and later



**Google Cloud SQL**

latest



**PostgreSQL**

9.0 and later (incl. Heroku & Amazon RDS)



**Vertica**

6.5 and later



**AWS Redshift**

latest



**DB2**

9.7 and later



**DB2 z/OS**

9.1 and later



**Derby**

10.8.2.2 and later



**H2**

1.2.137 and later



**Hsql**

1.8 and later



**SQLite**

3.7.2 and later



**solidDB**

6.5 and later

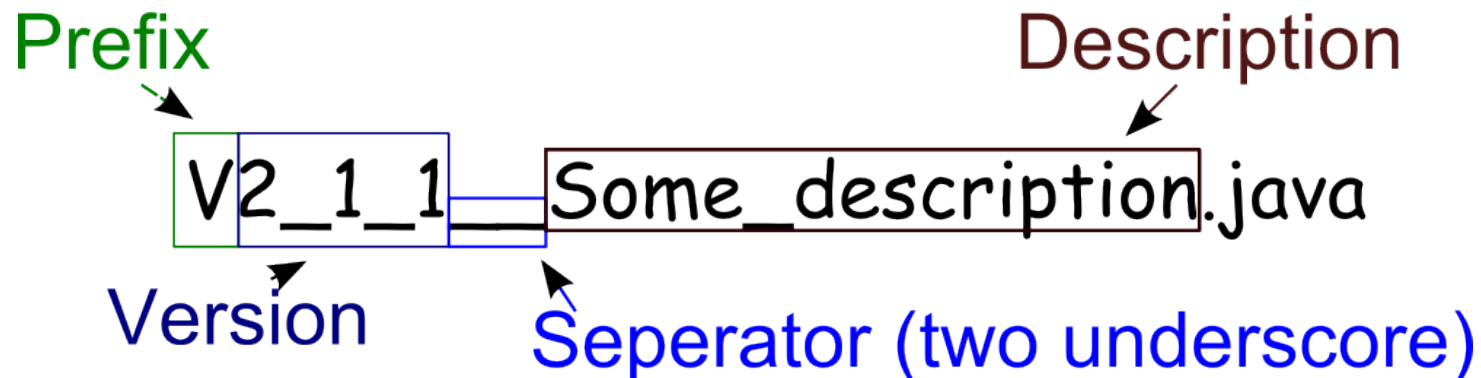


# Java Migration

- **Typische Anwendungsfälle**

- BLOB & CLOB Änderungen
- Fortgeschrittene Änderungen an Massendaten (Neuberechnungen, fortgeschrittene Formatsänderungen, ...)

- **Benennung der Java Klassen**



# Java Migration

## Beispiel

```
1 package db.migration;
2
3 import java.sql.Connection;
4 import java.sql.Statement;
5 import org.flywaydb.core.api.migration.jdbc.JdbcMigration;
6
7
8 public class V1_1_0__Insert_Data implements JdbcMigration {
9
10     @Override
11     public void migrate(Connection connection) throws Exception {
12         try (Statement statement = connection.createStatement()) {
13             statement.execute("Insert into person (first_name, last_name) Values ('Alice', 'Bob')");
14         }
15     }
16 }
17
18 }
19
```

# Java Migration

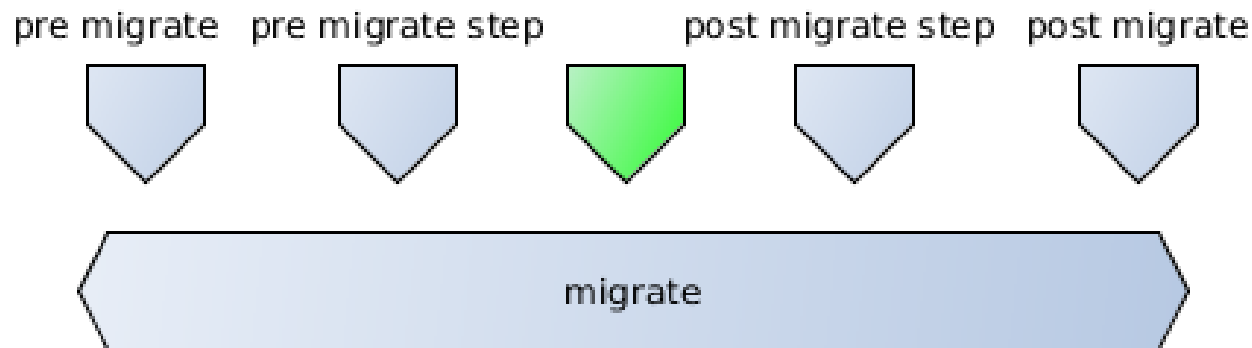
## Beispiel Spring Support

```
1 package db.migration;
2
3 import org.flywaydb.core.api.migration.spring.SpringJdbcMigration;
4 import org.springframework.jdbc.core.JdbcTemplate;
5
6
7 public class V1_2_0__Create_Table_With_Spring_Support implements SpringJdbcMigration {
8
9     @Override
10    public void migrate(JdbcTemplate jdbcTemplate) throws Exception {
11        jdbcTemplate.execute("Create table address (street Varchar(128), place Varchar(128))");
12    }
13
14 }
15
```

# Migration für Fortgeschrittene - Callbacks

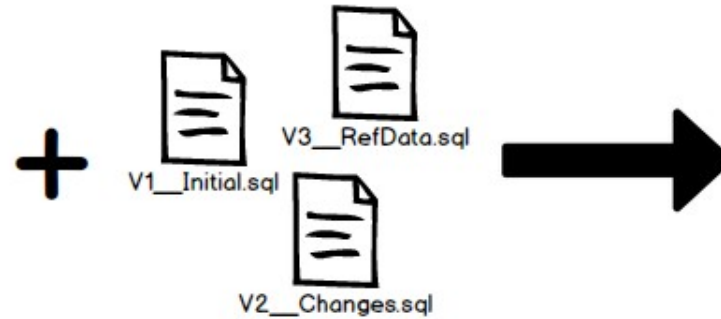
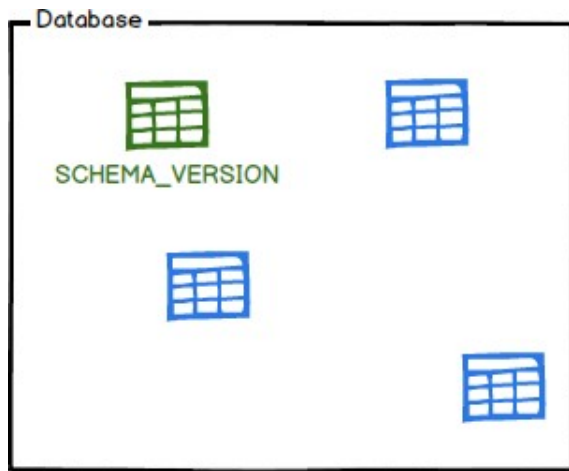
- **Typische Anwendungsfälle**
  - Stored Procedure Kompilierung
  - Materialized View Update

- **Flyway Lifecycle**



# Weitere Flyway Befehle

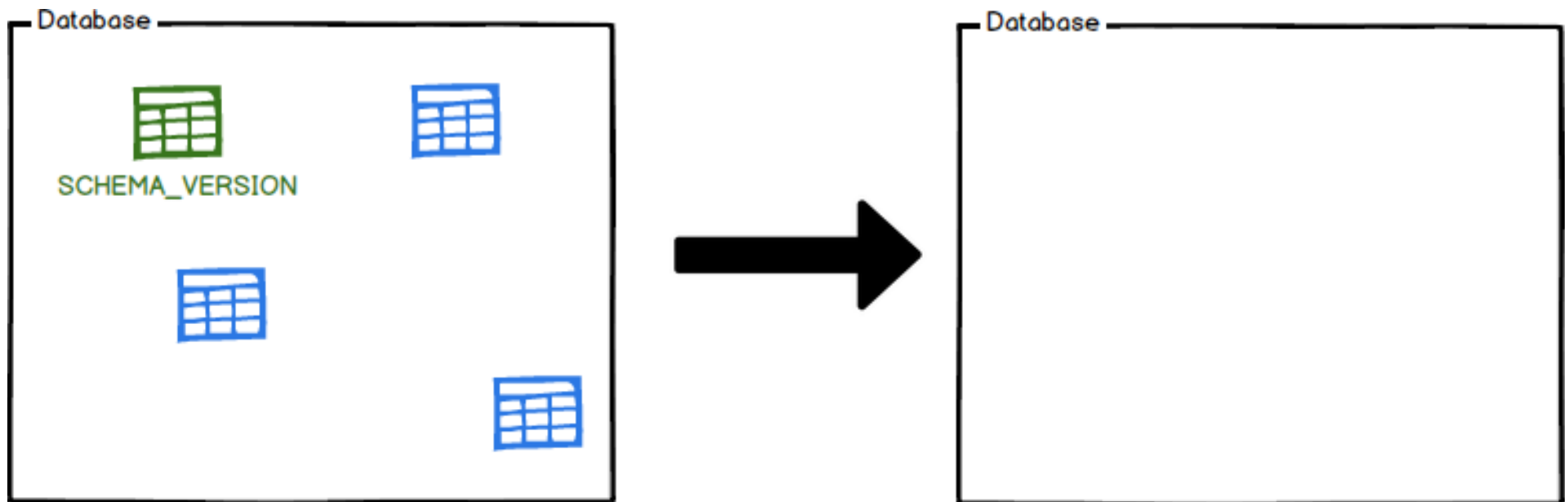
info



Version	Description	Installed on	State
1	Initial	2014-11-16 10:26:35	SUCCESS
2	Changes	2014-11-16 10:26:37	SUCCESS
3	RefData	2014-11-16 10:26:41	PENDING

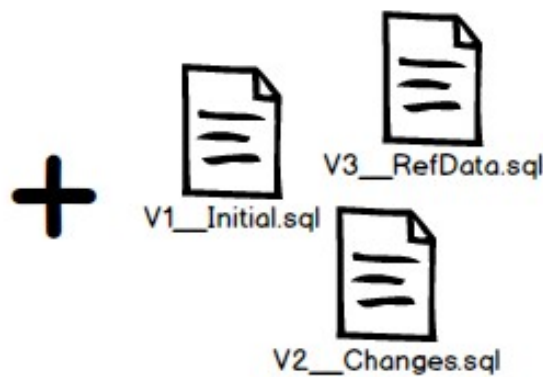
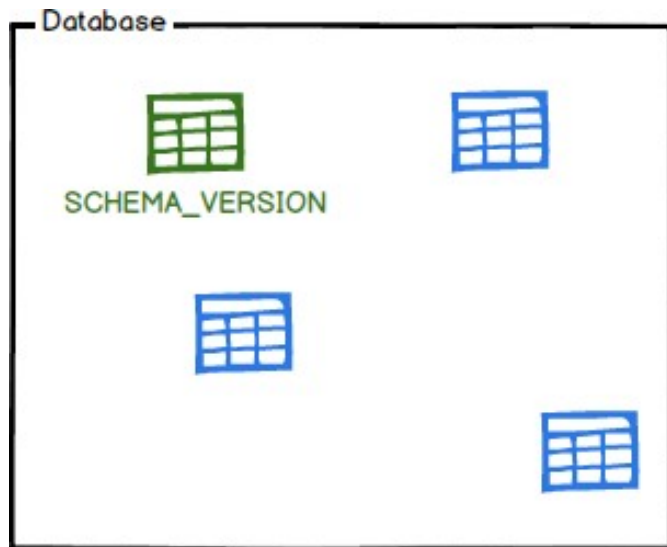
# Weitere Flyway Befehle

clean



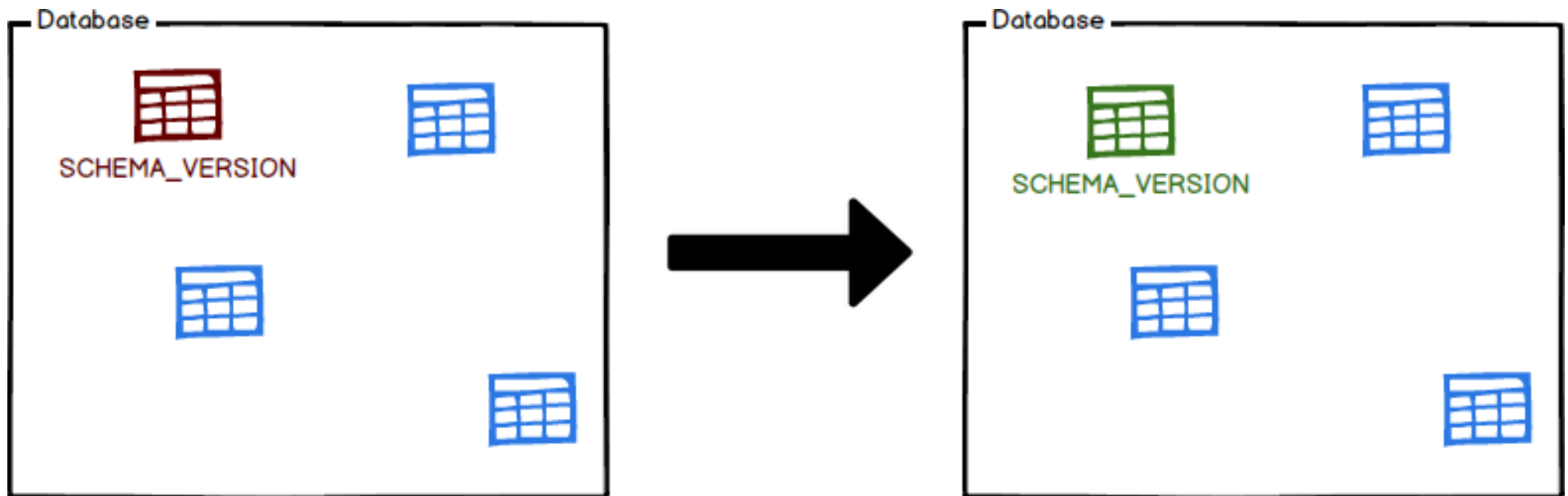
# Weitere Flyway Befehle

validate



# Weitere Flyway Befehle

repair





# Was kann Flyway nicht?

- Rollback Skripte aufrufen
- „Write once, run on many database vendors“

# Wie kann Flyway benutzt werden?

- **Flyway Clients:**

- Java API
- Maven Plugin
- Command-line Tool
- Gradle Plugin
- SBT Plugin
- Ant task

Demo

# Aufbau CDBI

- Behandle den Datenbank-Code wie einen ganz normalen Source-Code
  - Alle Datenbank Artefakte (DDL, DML, Konfigurationen, Testdaten, Stored Procedures, Functions etc) gehören ins VCS. ✓
  - Jede Änderung an den DB Artefakten wird getestet. ✓
- Jeder Entwickler hat seine eigene Datenbank / Testdatenbanken ähneln den Produktionsdatenbanken.
  - Automatisiertes Aufsetzen der Datenbank. ✓
- Änderungen an der Datenbank sind nachvollziehbar.
  - Historie der Änderungen ✓

Fallstricke

# Keine Instanz-spezifischen Daten

## Beispiel

1

2

3

4

```
GRANT SELECT, INSERT ON usermgm.* TO  
`technical-user`@'192.168.33.10' IDENTIFIED BY 'pA$$w0rt';
```

# Keine Instanz-spezifischen Daten

## Möglicher Lösungsansatz:

```
1  
2 GRANT SELECT, INSERT ON usermgm.* TO  
3 `technical-user`@'*' IDENTIFIED BY 'pA$$w0rt';  
4
```

---

- Zugriffskontrolle über eine Firewalls (iptables)

# Keine Instanz-spezifischen Daten

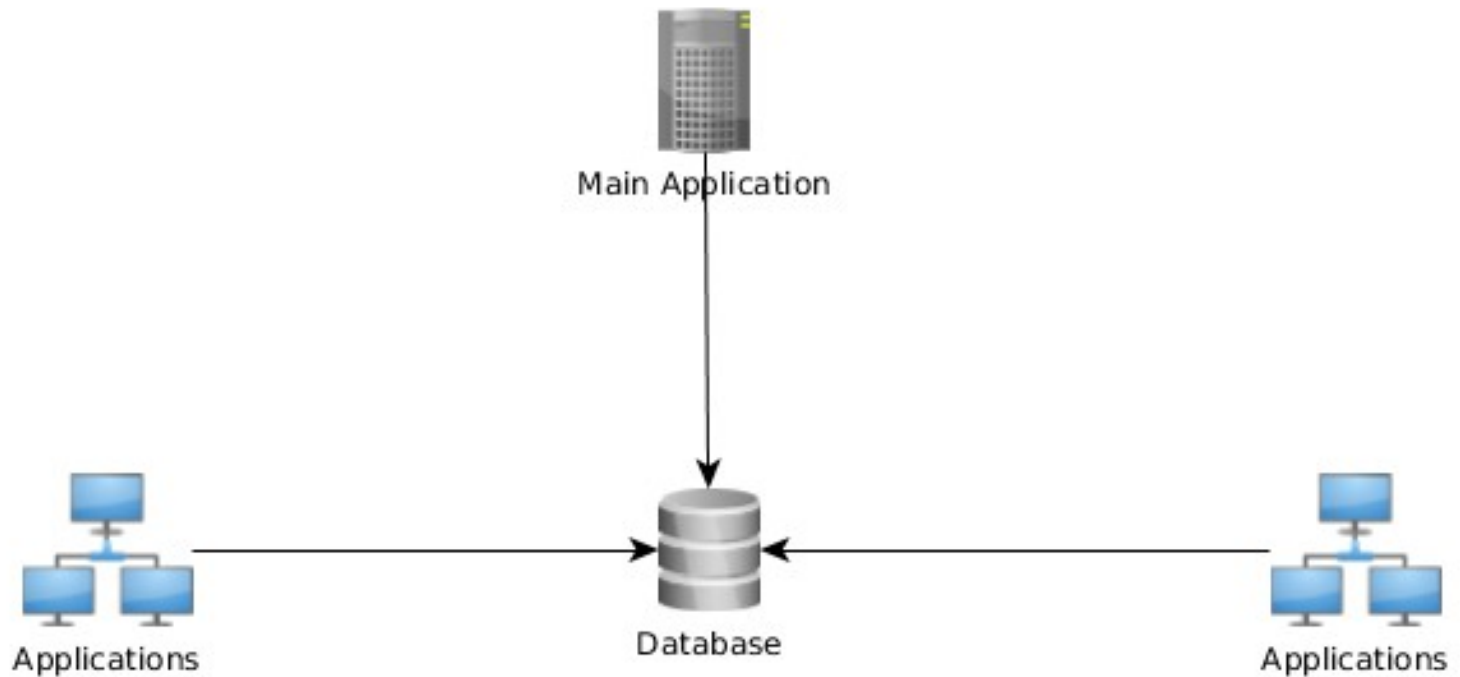
## Möglicher Lösungsansatz:

```
1 GRANT SELECT, INSERT ON usermgnt.* TO  
2 'technical-user' @ '${address}' By '${password}';  
3  
4
```



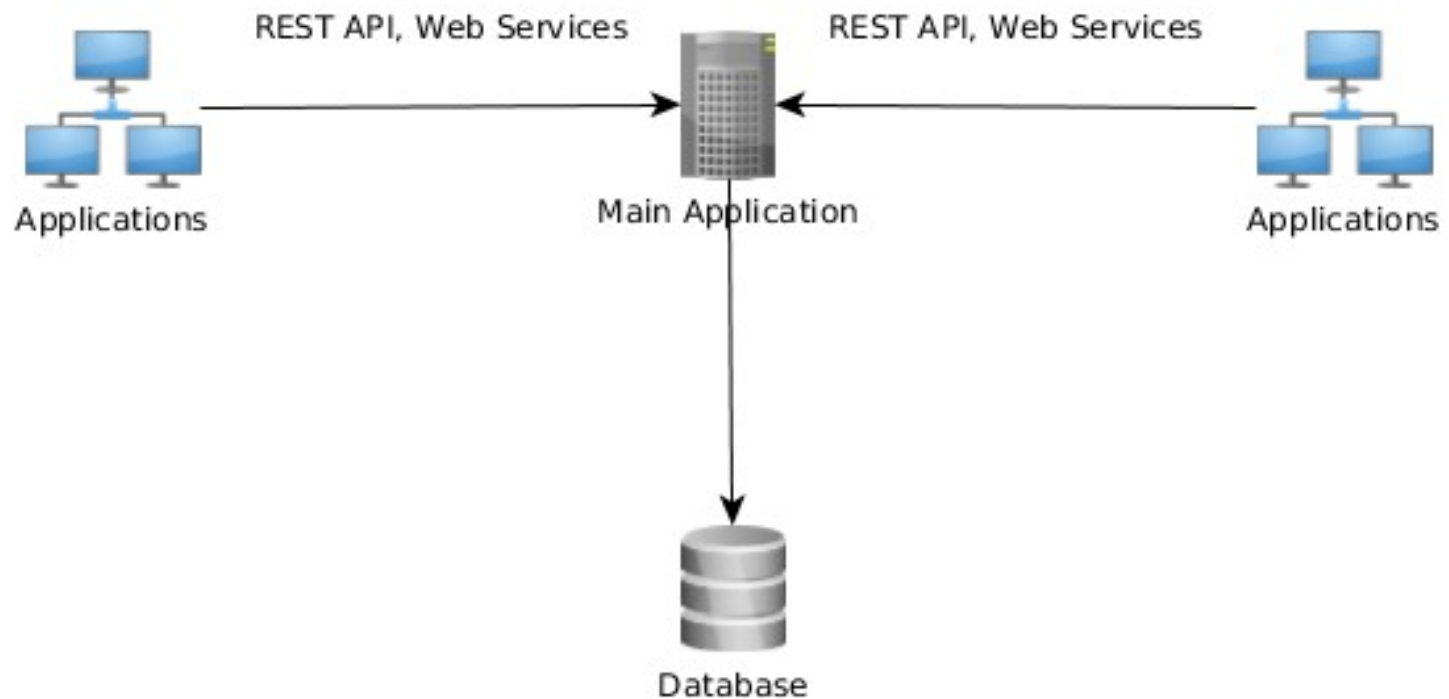
# Datenbank wird von mehreren Applikationen benutzt

**Ausgangslage :**



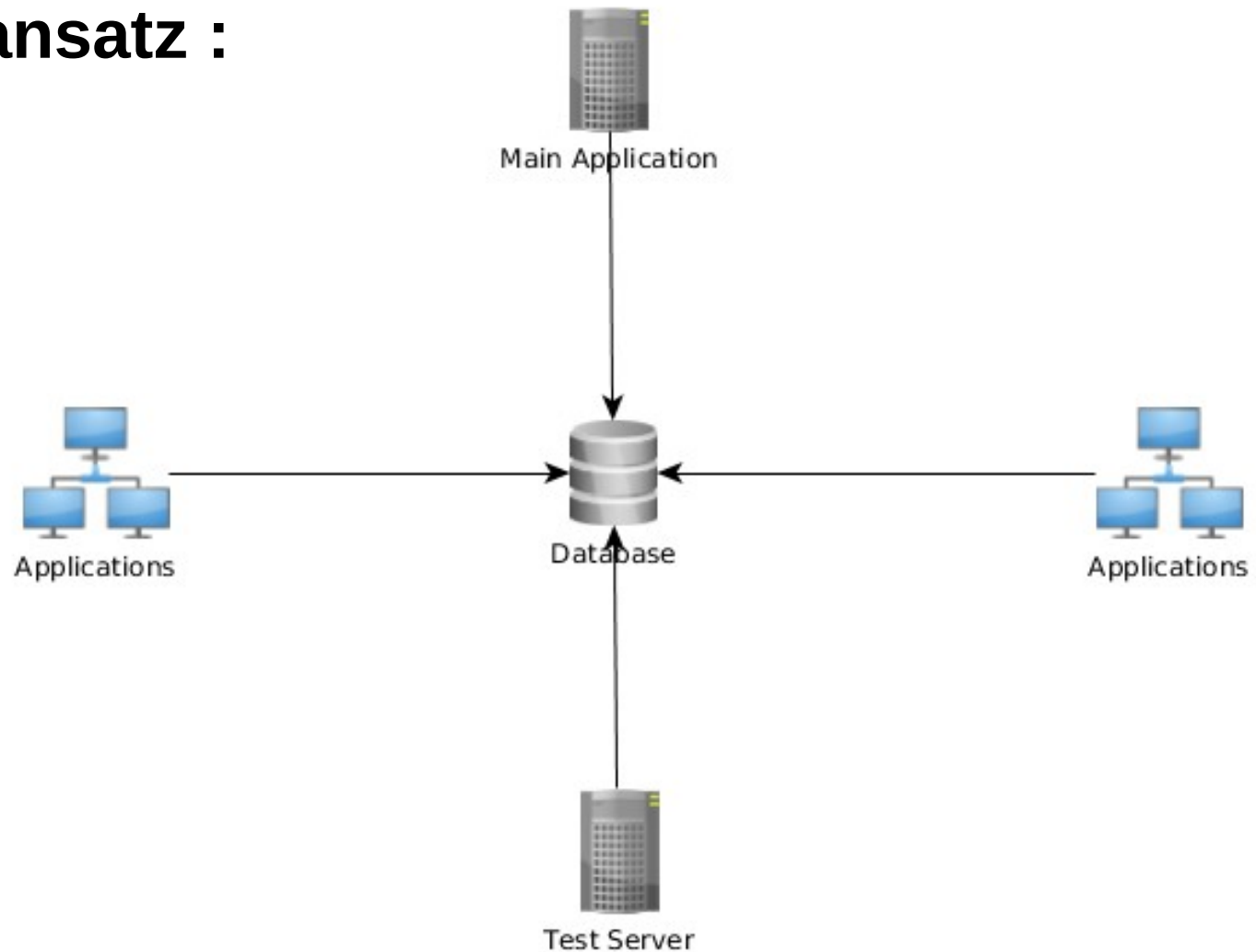
# Datenbank wird von mehreren Applikationen benutzt

**Lösungsansatz :**

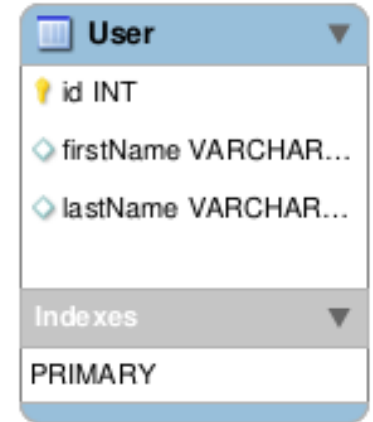
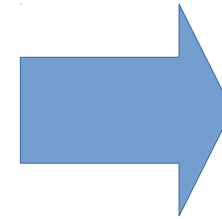
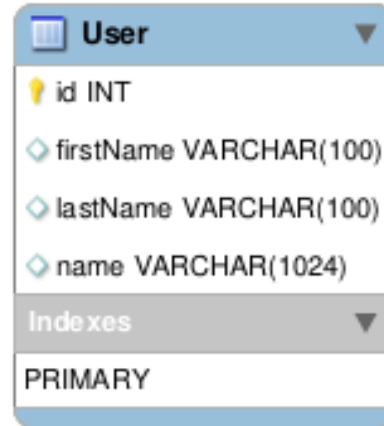
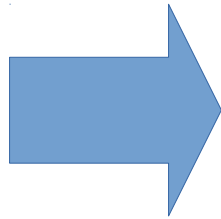
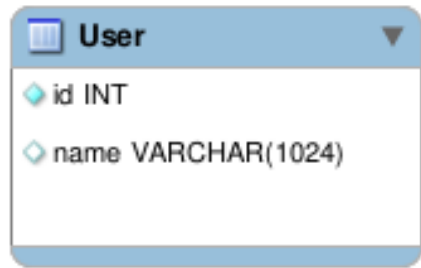


# Datenbank wird von mehreren Applikationen benutzt

**Lösungsansatz :**



# Datenbank wird von mehreren Applikationen benutzt



# Weitere Fallstricke (Auszug)

- Datenänderung dauern zu lange
- Datenlöschung
- Faktor Mensch
- ...

# Weitere Informationen

- Continuous Integration von Paul M. Duvall, Steve Matyas und Andrew Glover
- Refactoring Databases: Evolutionary Database Design von Scott J. Ambler und Pramodkumar J. Sadalage
- Flyway Documentation  
<http://flywaydb.org/documentation/migration/>  
<http://flywaydb.org/getstarted/>

# Fragen?

[info@sandra-parsick.de](mailto:info@sandra-parsick.de)  
@SandraParsick