Версионирование структуры баз данных на примере хранилища

Владислав Шишков
Системный инженер
vladislav.shishkov@lamoda.ru
09 декабря 2020 г.

мода на каждый день

lamoda

План

- Что было?
- Что хотели?
- Из чего выбирали?
- На чем остановились?
- Структура проекта
- Тестирование и деплой
- Подводные камни

Для чего это:

- версионирование кода для базы данных
- у нас много микросервисов с кучей баз

Что было?

Команды:

- 8 разработчиков DWH
- 20+ аналитиков для DWH в двух командах
- около 300 пользователей (Lamoda + GFG)
- десятки партнерских пользователей

В DWH:

- 700+ источников
- 30000+ таблиц плюс остальные объекты
- 85000+ процессов загрузки и обработки
- 7000+ отчетов, плюс bi, дашборды и ad-hoc
- деплоем до 20 задач в прод в день
- в каждой задаче происходит в среднем от 12 изменений структуры объектов (DDL)
- плюс логика для метаданных (DML)

Что было?

Базы данных:

- Oracle
- Vertica

VCS:

- SVN

Deploy:

- Python (for Oracle)
- manual (for Vertica)
- cron (autodeploy for DEV)
- Jira (trigger for cron-deploy)

lamoda

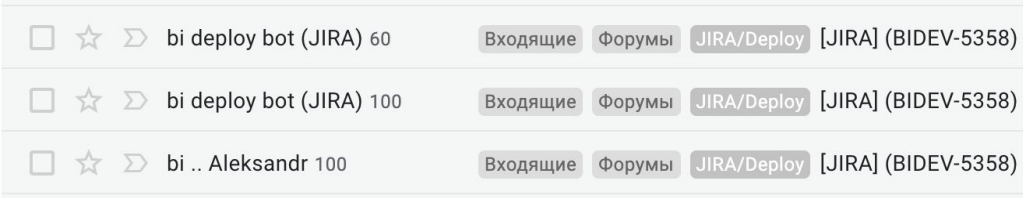
Что было?

Основные проблемы процесса разработки:

- сложно найти историю изменения объекта
- чуть сложнее найти задачу, в рамках истории объекта
- все эти проблемы связаны не только с разработчиками, но и с аналитиками
- аналитикам сложно “войти” в разработку, если нужны какие-то мелкие исправления

Что было?

- Множество писем при деплое
- Добавили буфер сообщений
- Сложность сопровождения самописного деплоя
- SVN - централизованная система управления версиями, упал сервер - встала работа
- Был откат SVN на 1 рабочий день и это грустно 😔
- Не было идемпотентности при выкате



Идемпотентность

Операция считается идемпотентной, если её многократное выполнение приводит к тому же результату, что и однократное выполнение.

Например, умножение на 1 — идемпотентная операция.

```
x * 1 == x * 1 * 1
```

Умножение на ноль — тоже идемпотентная операция.

```
x * 0 == x * 0 * 0
```

Идемпотентность в БД

В базах данных:

- DDL - Data Definition Language (язык описания данных):

  create, create or replace view, alter, drop, truncate
- DQL - Data Query Language (язык запросов данных)

  select
- DML - Data Manipulation Language (язык обработки данных):

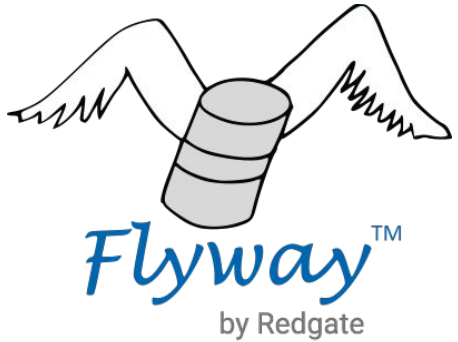
  insert, update, delete
- DCL - Data Control Language (язык управления базами данных):

  grant, revoke
- TCL - Transaction Control Language (язык управления транзакциями)

  commit, rollback, savepoint, set transaction

Из чего выбирали?

- Flyway (flywaydb.org)



- Sqitch (sqitch.org)

- Liquibase (liquibase.org)

Что хотели?

Идемпотентность в Flyway и Liquibase практически идентичны - через схему и таблицу в базе.
Удобно, когда прям в базе сохраняется состояние деплоя, можно в одном проекте держать разные стейджы с базами и соответственно с разным состоянием.

Идемпотентность в Sqitch реализована через аналог Git репы в виде дерева хешей.
Сохраняется локально. Возможны проблемы при расхождении проекта Sqitch с откатанной базой.

Что хотели?

ODBC и JDBC подключение

ODBC поддерживают все три инструмента.

Flyway и Liquibase написаны на Java, поддержка JDBC есть из коробки.

Sqitch написан на perl, поэтому только ODBC.

Специфика баз Oracle и Vertica:

- для ODBC нужно устанавливать драйвера и приложения, усложняет сборку и настройку в контейнерах
- Vertica по разному работает с балансировщиком в зависимости от драйвера
- у Vertica есть дополнительная функция MARS только для JDBC

Что хотели?

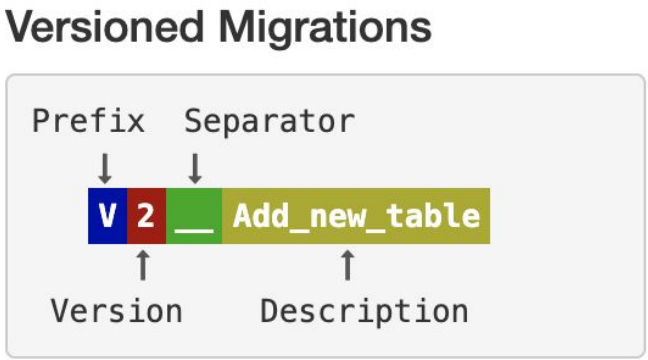
DDL / DML

Все три инструмента поддерживают работу как с DDL, так и с DML, но:

- в Liquibase можно использовать результат DML в условиях или тестах
- в Sqitch есть verify для тестирования на sql

Структура проекта:

- в Flyway структура папок фиксирована, названия файлов имеют логику…
- в Sqitch структура папок фиксирована
- в Liquibase можно использовать любую структуру



Что хотели?

Raw SQL / DSL (domain-specific language)

Все три инструмента поддерживают работу на sql, но только Liquibase имеет встроенный DSL язык, для унифицированного обращения к объектам базы

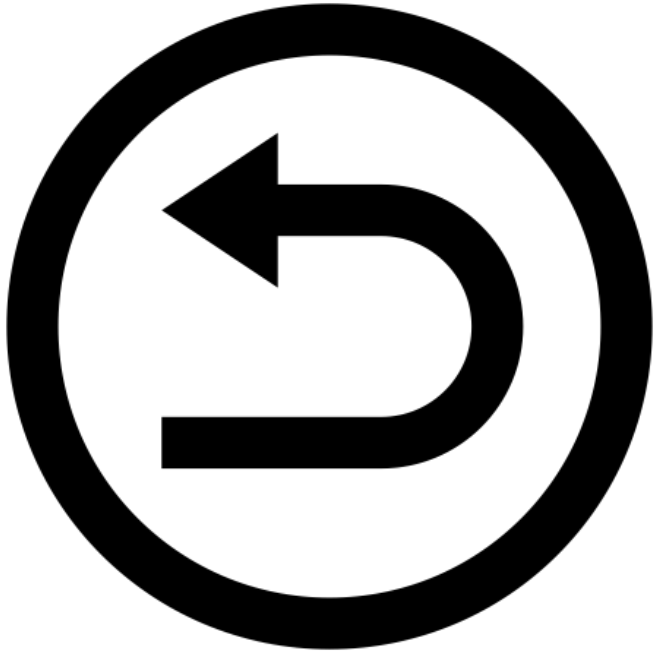
```
<changeSet  id="1"  author="nvoxland">
    <createTable  tableName="person">
        <column  name="id"  type="int"  autoIncrement="true">
            <constraints  primaryKey="true"  nullable="false"/>
        </column>
        <column  name="firstname"  type="varchar(50)"/>
        <column  name="lastname"  type="varchar(50)">
            <constraints  nullable="false"/>
        </column>
        <column  name="state"  type="char(2)"/>
    </createTable>
</changeSet>
```

Что хотели?

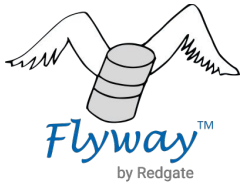
Dry run - холостой прогон, проверка и понимание, что произойдет

Undo - поддержка откатов

Присутствует во всех трех инструментах, но в Flyway только в платной поддержке...



Из чего выбирали?

Требования		SQITCH Liquibase
ODBC/JDBC connection	✓ ✓	✓ ✗ ✓ ✓
Идемпотентность	✓	✓ ✓
Raw SQL / DSL (domain-specific language)	✓ ✗	✓ ✗ ✓ ✓
DDL / DML	✓ ✓	✓ ✓ ✓ ✓
Dry run / Undo	only PRO	✓ ✓ ✓ ✓

Liquibase

Основные термины:

- changelog files: SQL, XML, JSON, YAML, Other
- changesets - единица изменения
- Change types - типы изменений
- Preconditions - условия
- Contexts / Labels - логика обработки и выполнения

```xml
<?xml version="1.0" encoding="UTF-8"?>

<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:pro="http://www.liquibase.org/xml/ns/pro"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog http://www.liquibase.org
      http://www.liquibase.org/xml/ns/pro http://www.liquibase.org/xml/ns/pro/liquibase-pr
    <changeSet  id="1"  author="bob">
        <comment>A sample change log</comment>
        <createTable/>
    </changeSet>
    <changeSet  id="2"  author="bob"  runAlways="true">
        <alterTable/>
    </changeSet>
    <changeSet  id="3"  author="alice"  failOnError="false"  dbms="oracle">
        <alterTable/>
    </changeSet>
    <changeSet  id="4"  author="alice"  failOnError="false"  dbms="!oracle">
        <alterTable/>
    </changeSet>

</databaseChangeLog>
```

Liquibase work logic

- DATABASECHANGELOG
- DATABASECHANGELOGLOCK
- id/author/filepath - unique id
- MD5sum
- runOnChange
- rollback for any changeset

```
<changeSet  id="1"  author="bob">
    <createTable  tableName="testTable">
    <rollback>
        <dropTable  tableName="testTable"/>
    </rollback>
</changeSet>
```

```
<changeSet  id="2"  author="bob">
    <dropTable  tableName="testTable"/>
    <rollback  changeSetId="1"  changeSetAuthor="bob"/>
</changeSet>
```

Liquibase commands

Основные команды:

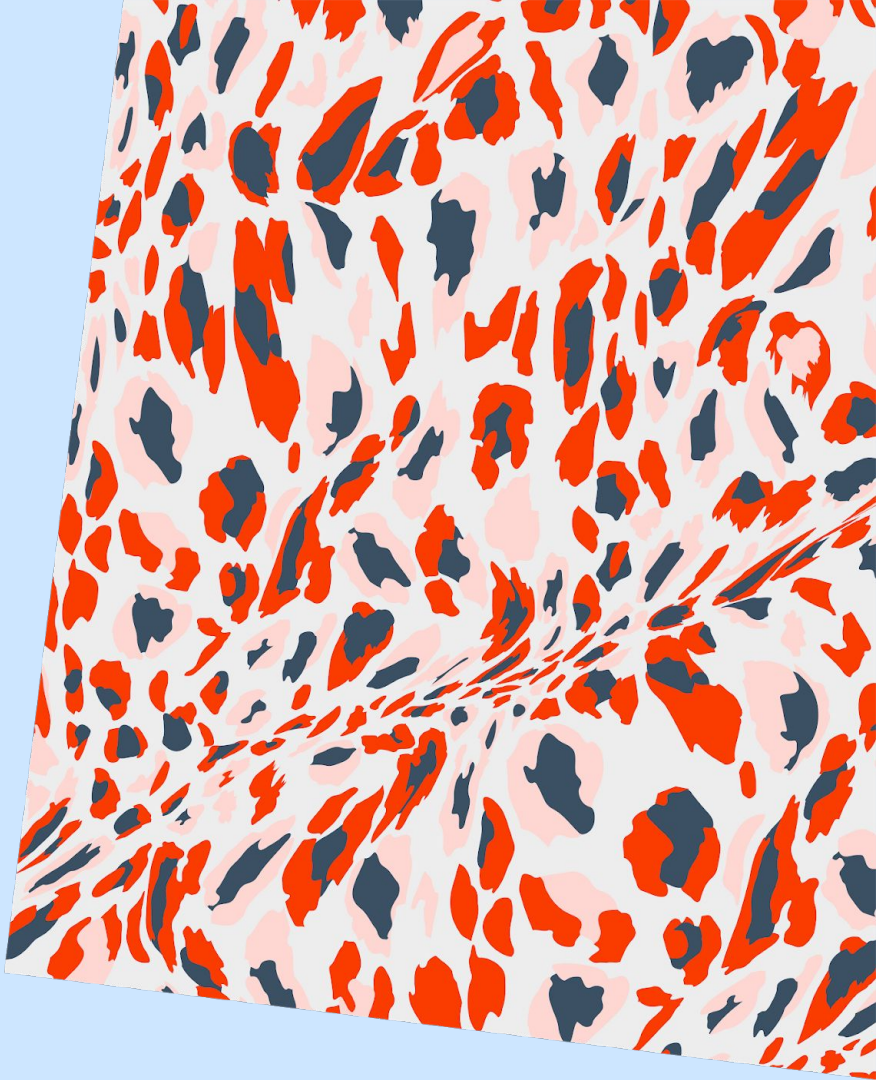
- update - применение миграции
- updateSQL - dry run
- validate - проверка changelog на ошибки
- changelogSync - фиксация проекта в базе
- generateChangeLog - генерация DSL на основе существующей структуры базы данных

Основные поддерживаемые СУБД:

- MySQL / MariaDB
- PostgreSQL
- Oracle
- MS SQL Server
- Vertica
- and more 14 DBMS
- custom and plugins

Паттерны структуры проекта

- Миграционные патчи
- Объектная модель
- Гибрид



lamoda

Миграционные патчи

Структура проекта:

Project
- Migrations
  - <task_id1>
    - changelog.xml
    - other1.sql
  - <task_id2>
    - changelog.xml
    - other2.sql
- main.xml

Плюсы:

- последовательность - деплой/откат
- работа в рамках патча/задачи
- наименее проблемная структура проекта

Минусы:

- diff - не видно истории в рамках объекта

Миграционные патчи

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<databaseChangeLog
        xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
        xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchange
        http://www.liquibase.org/xml/ns/dbchangelog-ext http://www.l
    <include file="_MIGRATIONS/_init/init.xml"/>
    <include file="_MIGRATIONS/BIDEV-8246/BIDEV-8246.xml"/>
    <include file="_MIGRATIONS/BIDEV-8328/BIDEV-8328.xml"/>
    <include file="_MIGRATIONS/BIDEV-8333/BIDEV-8333.xml"/>
    <include file="_MIGRATIONS/BIDEV-8191/BIDEV-8191.xml"/>
    <include file="_MIGRATIONS/BIDEV-8185/BIDEV-8185.xml"/>
    <include file="_MIGRATIONS/BIDEV-8198/BIDEV-8198.xml"/>
    <include file="_MIGRATIONS/BIDEV-8285/BIDEV-8285.xml"/>
    <include file="_MIGRATIONS/BIDEV-7949/BIDEV-7949.xml"/>
    <include file="_MIGRATIONS/BIDEV-8313/BIDEV-8313.xml"/>
    <include file="_MIGRATIONS/BIDEV-8176/BIDEV-8176.xml"/>
    <include file="_MIGRATIONS/BIDEV-8357/BIDEV-8357.xml"/>
    <include file="_MIGRATIONS/BIDEV-8248/BIDEV-8248.xml"/>
    <include file="_MIGRATIONS/BIDEV-8322/BIDEV-8322.xml"/>
    <include file="_MIGRATIONS/BIDEV-8337/BIDEV-8337.xml"/>
    <include file="_MIGRATIONS/BIDEV-7608/BIDEV-7608.xml"/>
    <include file="_MIGRATIONS/BIDEV-8264/BIDEV-8264.xml"/>
    <include file="_MIGRATIONS/BIDEV-8360/BIDEV-8360.xml"/>
</databaseChangeLog>
```

Объектная модель

Структура проекта:

Project
- DB_name
  - <schema_name>
    - tables
      - <table_name>.sql
    - views
      - <view_name>.sql
  - <schema_name>.xml
- main.xml

Плюсы:

- diff - видно историю по объектам
- разработка в рамках объектов

Минусы:

- выстраивание последовательности в рамках объектов
- деплой от текущего состояния до следующего

Объектная модель

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
                   xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchange
                                       http://www.liquibase.org/xml/ns/dbchange
    <include file="DWH/ROCKET_DWH_BL_STA/ROCKET_DWH_BL_STA.xml"/>
    <include file="DWH/DWH_BL/DWH_BL.xml"/>
    <include file="DWH/ROCKET_DWH_BL/ROCKET_DWH_BL.xml"/>
    <include file="DWH/_CREATE/create.xml"/>
</databaseChangeLog>
```

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
                   xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchange
                                       http://www.liquibase.org/xml/ns/dbchange
    <includeAll path="DWH/ROCKET_DWH_BL/schemas/"/>
    <includeAll path="DWH/ROCKET_DWH_BL/tables/"/>
    <includeAll path="DWH/ROCKET_DWH_BL/views/"/>
</databaseChangeLog>
```

Гибридная модель

Структура проекта:

```
Project
    Migrations
        <task_id1>
            changelog.xml
            other1.sql
    DB_name
        <schema_name>
            tables
                <table>
                    <table_name>.sql
                    <task_id>.sql
            views
                <view_name>.sql
        <schema_name>.xml
    main.xml
```

Плюсы:

- diff - видно историю по объектам
- разработка в рамках объектов
- гибкость в рамках миграционных патчей
- проще откат, чем в объектной модели

Минусы:

- деплой от текущего состояния до следующего
- последовательность в рамках объектов и патчей
- сложнее с контролем выполнения
- при откате остаются проблемы с идемпотентными объектами (но решаемо со стороны VCS)

Гибридная модель

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<databaseChangeLog
        xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
        xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchange
        http://www.liquibase.org/xml/ns/dbchangelog-ext http://www.l
    <include file="objects_pre.xml"/>
    <include file="migrations.xml"/>
    <include file="objects_post.xml"/>
    <include file="regress_tests.xml"/>
</databaseChangeLog>
```

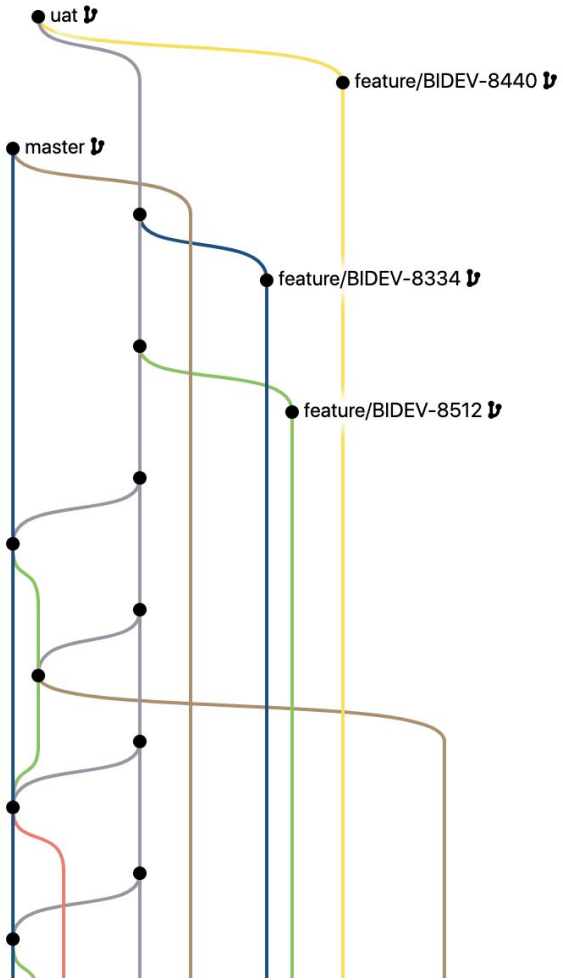
```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<databaseChangeLog
        xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ext="http://www.liquibase.org/xml/ns/dbchangelog-ext"
        xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangel
        http://www.liquibase.org/xml/ns/dbchangelog-ext http://www.li

    <include file="_OBJECTS/ROCKET_DWH_DL/tables/DG_APX_ZONE_NAME_DIC
    <include file="_OBJECTS/ROCKET_DWH_DL/tables/DL_APX_ZONE_NAME_DIC
    <include file="_OBJECTS/ROCKET_DWH_DL/tables/ERR_DG_APX_ZONE_NAME
    <include file="_OBJECTS/ROCKET_DWH_DL/tables/STA_APX_ZONE_NAME_DI
    <include file="_OBJECTS/ROCKET_DWH_DL/tables/TMP_DG_APX_ZONE_NAME
    <include file="_OBJECTS/ROCKET_DWH_DL/views/V_DG_APX_ZONE_NAME_DI
    <include file="_OBJECTS/ROCKET_DWH_DL/packages/PCT_APX_ZONE_NAME_

    <include file="_MIGRATIONS/BIDEV-8185/001_ml.sql"/>

</databaseChangeLog>
```

Тестирование и деплой

lamoda

Workflow проекта в git

- Фиксированная ветка для каждого стейджа
- Фича-ветки всегда начинаются и актуализируются от прод ветки (в нашем случае от master)
- Фича-ветки мержаться в стейдж-ветки
- По необходимости, запрет выката на прод без выката на другой стейдж
- Актуализация стейдж-веток после мержа на прод



lamoda

Тестирование проекта

При разработке:

- updateSQL - просмотр фактических sql запросов

На этапе интеграции/сборки:

- валидация проекта со стейдж настройками (для фича-ветки настройки с прода)
- если есть возможность, поднятие контейнера, выкат текущего прода, накат фича-ветки
- и конечно же откат

Деплой проекта

На этапе доставки:

- logLevel=info
- валидация проекта
- доставка
- запуск регресс тестов
- вывод результата тестов в лог
- оповещения в слак о деплое

Подводные камни Liquibase

lamoda

Подводные камни liquibase

1. Смена loglevel привела к неожиданному результату (но это не точно):
    - output “дублируется” при logLevel=info
    - preConditions и значения параметров в onFail/onError

      внимательно читаем документацию для WARN и CONTINUE
2. validate не дает гарантии работы **всей** структуры

Свой validate

- XML not included
- XML FILES not found (*)
- SQL not included
- SQL FILES not found (*)
- XML duplicated
- SQL duplicated

```
25-Nov-2020 18:02:35  ============ XML not included ============
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7302/BIDEV-7302.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7608/BIDEV-7608.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7949/BIDEV-7949.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8176/BIDEV-8176.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8185/BIDEV-8185.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8191/BIDEV-8191.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8198/BIDEV-8198.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8246/BIDEV-8246.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8248/BIDEV-8248.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8264/BIDEV-8264.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8285/BIDEV-8285.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8286/BIDEV-8286.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8313/BIDEV-8313.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8322/BIDEV-8322.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8328/BIDEV-8328.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8333/BIDEV-8333.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8337/BIDEV-8337.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8357/BIDEV-8357.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8360/BIDEV-8360.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8377/BIDEV-8377.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8383/BIDEV-8383.xml
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8385/BIDEV-8385.xml
25-Nov-2020 18:02:35  ============ SQL not included ============
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7302/001_drop_service_level_group.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7302/002_ml_service_level_group.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7302/003_drop_shipping_method.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7302/004_delete_ml_xdc_glossary.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7608/010_BIDEV-7608_meta.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7608/011_BIDEV-7608_wf.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-7949/010_BIDEV-7949_test_suite_settings.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8176/001_dg_meta.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8176/002_il_meta.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8185/001_ml.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8191/03_8191_meta.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8198/8198_disable_old_wf.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8198/8198_fw_meta.sql
25-Nov-2020 18:02:35  _MIGRATIONS/BIDEV-8198/8198_wf_meta.sql
```

lamoda

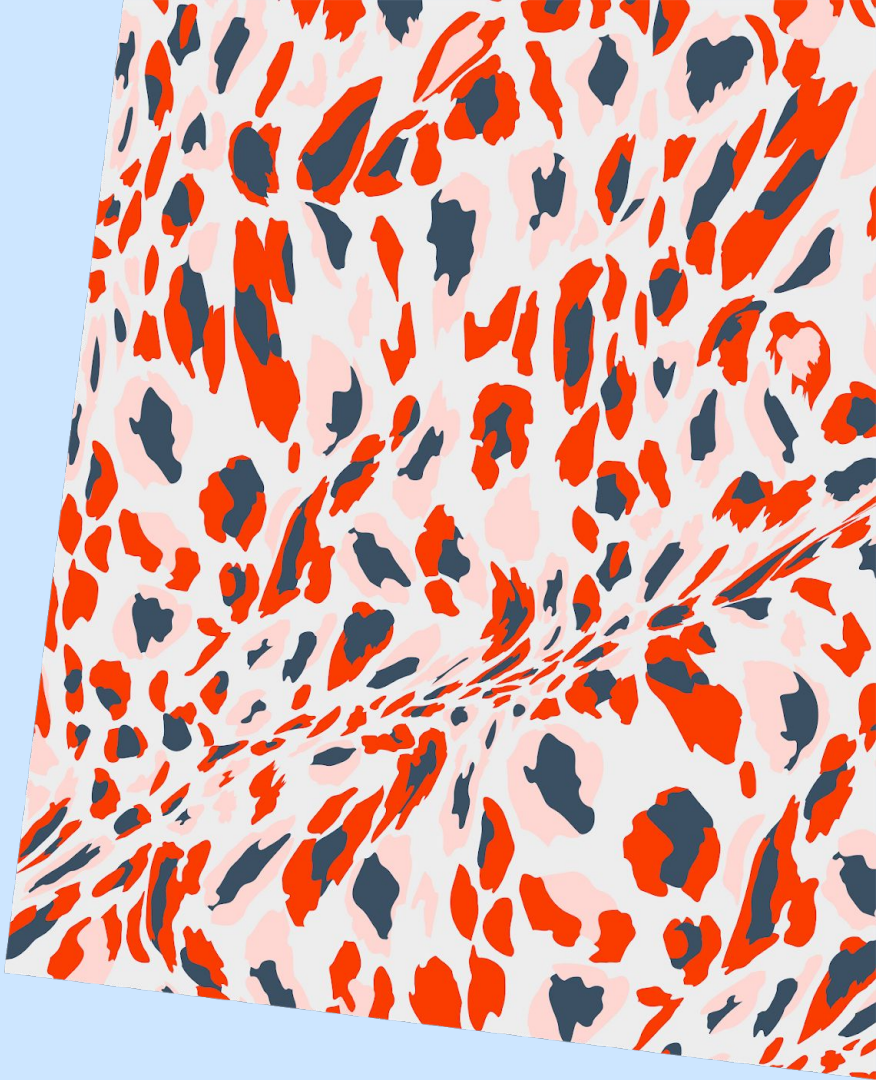
Результат

- Код задокументирован
- И даже полноценно историчен

  (в зависимости от структуры)
- Валидация синтаксиса
- Валидация и тесты
- Автоматизирован процесс

Вывод

Версионирование структуры баз данных удобнее с инструментами



lamoda

Владислав Шишков
Системный инженер
vladislav.shishkov@lamoda.ru
09 декабря 2020 г.

Вопросы?

мода на каждый день

lamoda