

# ДЕПЛОЙ И ОТКАТ

## ПРИЛОЖЕНИЯ И МИГРАЦИЙ БД

# ПЛАН РАССКАЗА

- Постановка задачи.
- Прямые и обратные SQL операции.
- Пример реализации: делпой и откат приложения в кластере k8s.
- Ограничения решения.
- Как настроить liquibase для применения отката миграций.

# ПОСТАНОВКА ЗАДАЧИ

Нам надо обновлять PSI контур из **feature**-веток и иметь возможность откатывать до ветки **develop**, чтобы потом накатить другую ветку.

```
$ helm rollback calm-core "$REVISION" --wait
```

А как откатывать миграции базы данных?!

# ПРЯМЫЕ И ОБРАТНЫЕ SQL ОПЕРАЦИИ

## ПРЯМАЯ ОПЕРАЦИЯ

## ОБРАТНАЯ ОПЕРАЦИЯ

CREATE TABLE  
(VIEW, INDEX, FUNCTION, TRIGGER, SEQUENCE)

DROP

ADD COLUMN

DROP COLUMN

ALTER COLUMN

ALTER COLUMN

ALTER ... RENAME TO ...

ALTER ... RENAME TO ...

DROP TABLE  
(VIEW, FUNCTION, TRIGGER, SEQUENCE)

CREATE

# ПРЯМЫЕ И ОБРАТНЫЕ SQL ОПЕРАЦИИ

ПРЯМАЯ ОПЕРАЦИЯ	ОБРАТНАЯ ОПЕРАЦИЯ
DROP COLUMN (NOT NULL)	ADD COLUMN UPDATE ... SET ... ALTER COLUMN SET NOT NULL
DROP CONSTRAINT	DELETE ... (records, which not meet constraint) CREATE CONSTRAINT
INSERT	DELETE (can leave unmodified, if doesn't affect process)
UPDATE	UPDATE (can leave unmodified, if doesn't affect process)
DELETE, TRUNCATE	INSERT (if possible)

# ПРИМЕР

Просто приложение с 2 методами:

- Версия приложения и сборки  
**GET** <http://localhost:8080/api/v1/version>
- Запрос на получение пользователей из БД  
**GET** <http://localhost:8080/api/v1/users>

6 миграций:

1. Создание таблицы users.
2. Изменение типа столбца id на IDENTITY.
3. Создание таблицы address и FK от users → address.
4. Изменение поля status на enum.
5. Изменение таблицы users на партиционирование.
6. Увеличение размера поля login до 80 символов.

# ОГРАНИЧЕНИЯ

- Не всем миграции можно откатить:  
**INSERT, UPDATE, DELETE**
- Откат миграций требуется **тщательно** проверять отдельно:  
**ALTER TABLE ... RENAME TO ...** изменяет  
кроме имени таблицы еще и зависимые  
**CONSTRAINT, INDEX, SEQUENCE.**
- Требуется ставить связывать версию миграции и версию деплоя.

# НАСТРОЙКА LIQUIBASE

- В конец каждой миграции добавляем tag:  

```
<changeSet id="3" author="aromanow" labels="v1.0">  
    <tagDatabase tag="v1.0"/>  
</changeSet>
```
- Добавляем проверку отката миграций  
(необходимое, но не достаточное условие)  

```
spring:  
  liquibase:  
    change-log: classpath:db/liquibase/changelog.xml  
    test-rollback-on-update: true
```



# НАСТРОЙКА LIQUIBASE

- Liquibase выполняет откат по имени миграций (filename).

Полное имя миграций при запуске из Spring Boot и из Docker отличается (при запуске через Spring Boot filename берет полный путь внутри jar). Чтобы исправить это, требуется явно прописать имя миграции:

```
<databaseChangeLog
  logicalFilePath="v1.0_CreateUserTable.xml"
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
    http://www.liquibase.org/xml/ns/dbchangelog/4.20.xsd">
  <changeSet id="1" author="aromanow" labels="v1.0">
    ...
  </changeSet>
</databaseChangeLog>
```

# НАСТРОЙКА LIQUIBASE

- При откате релиза через Helm используется номер ревизии. Чтобы связать его с именем версии, нужно при деплое указать версию миграции как description:

```
$ helm upgrade \  
  migration-application \  
  romanow/java-service \  
  --values=k8s/migration-application/values.yaml \  
  --set image.tag="$IMAGE_VERSION" \  
  --description "$IMAGE_VERSION" \  
  --install \  
  --wait
```

# ВОПРОСЫ