# Apache ShardingSphere 数据脱敏

作者：TravelEngineers
原文链接：https://ld246.com/article/1567132596332
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

# 写在前面

由于现在SpringBoot很火，因此本文主要使用SpringBoot 2.x进行相关介绍。

Apache ShardingSphere Github:

https://github.com/apache/incubator-shardingsphere

ShardingSphere说明文档：https://shardingsphere.apache.org/document/current/cn/overview/

# 需求场景分析

对于数据脱敏的需求，在现实的业务场景中一般分为两种情况：

1. 新业务上线，安全部门规定需将涉及用户敏感信息，例如银行、手机号码等进行加密后存储到数据，在使用的时候再进行解密处理。因为是全新系统，因而没有存量数据清洗问题，所以实现相对简单。

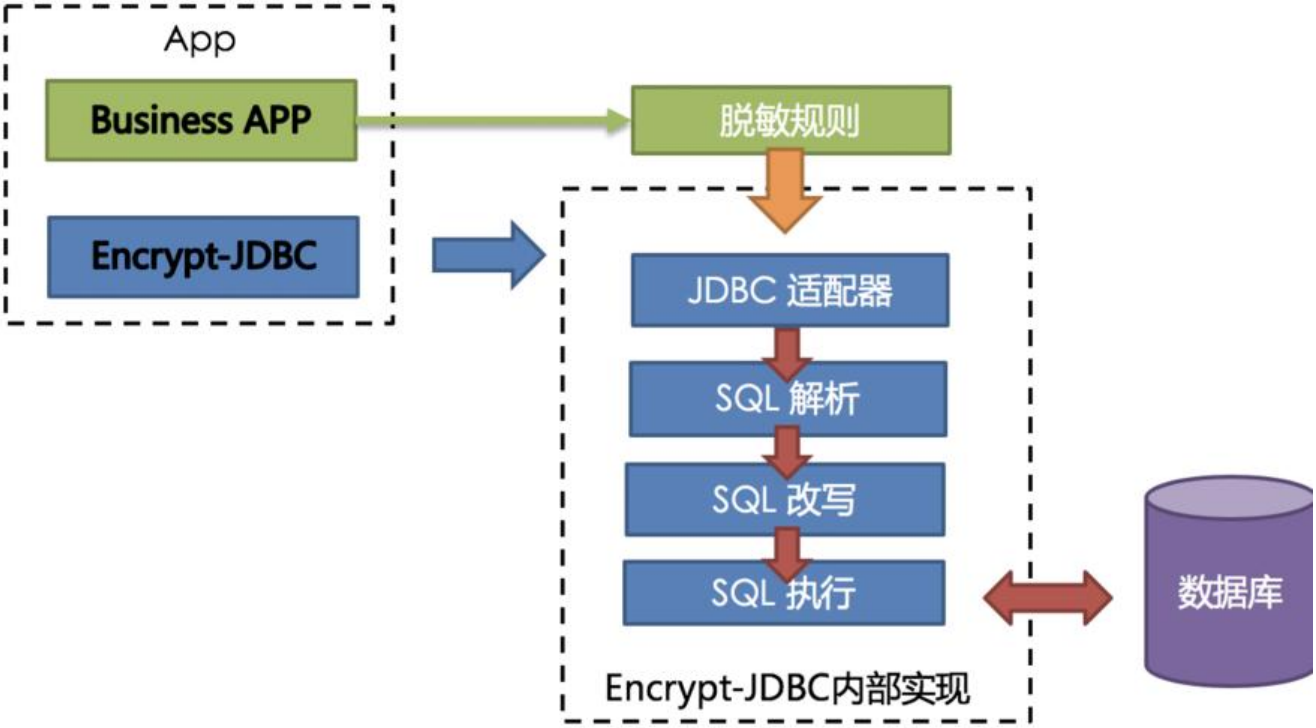
2. 已上线业务，之前一直将明文存储在数据库中。相关部门突然需要对已上线业务进行脱敏整改。这场景一般需要处理三个问题：

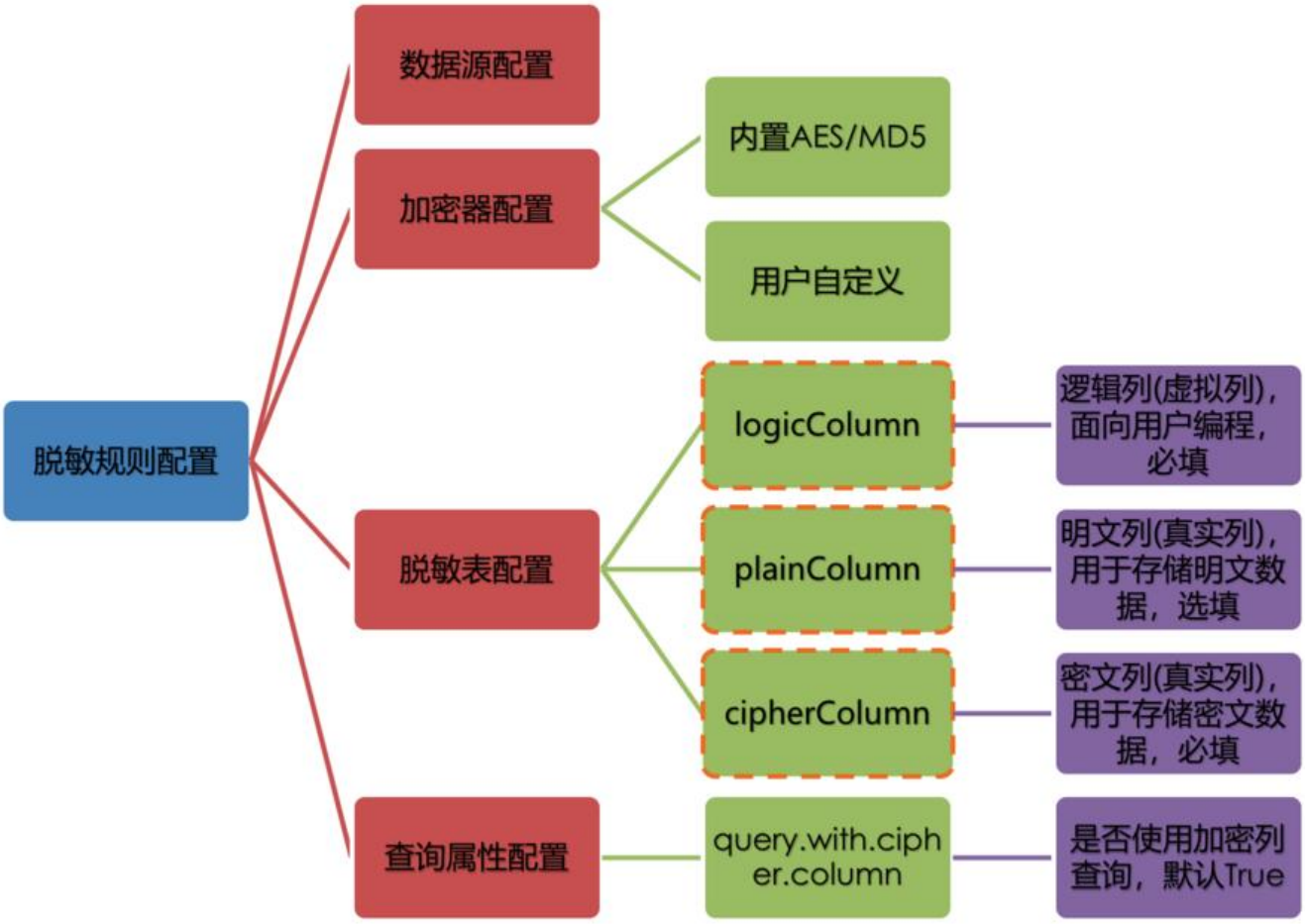
a) 历史数据需要如何进行脱敏处理，即洗数。

b) 如何能在不改动业务SQL和逻辑情况下，将新增数据进行脱敏处理，并存储到数据库；在使用时，进行解密取出。

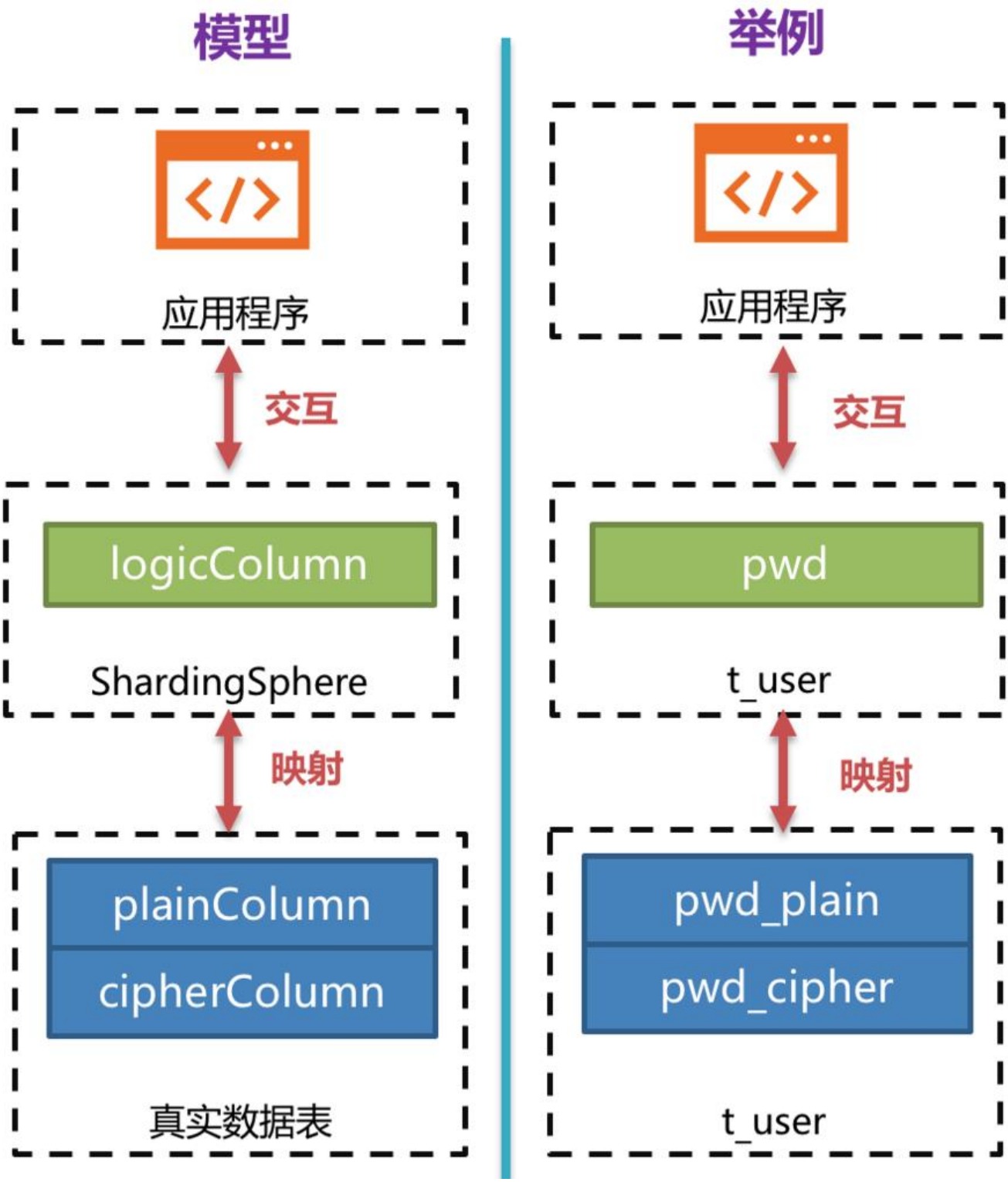
c) 如何较为安全、无缝、透明化地实现业务系统在明文与密文数据间的迁移。

# 处理流程详解

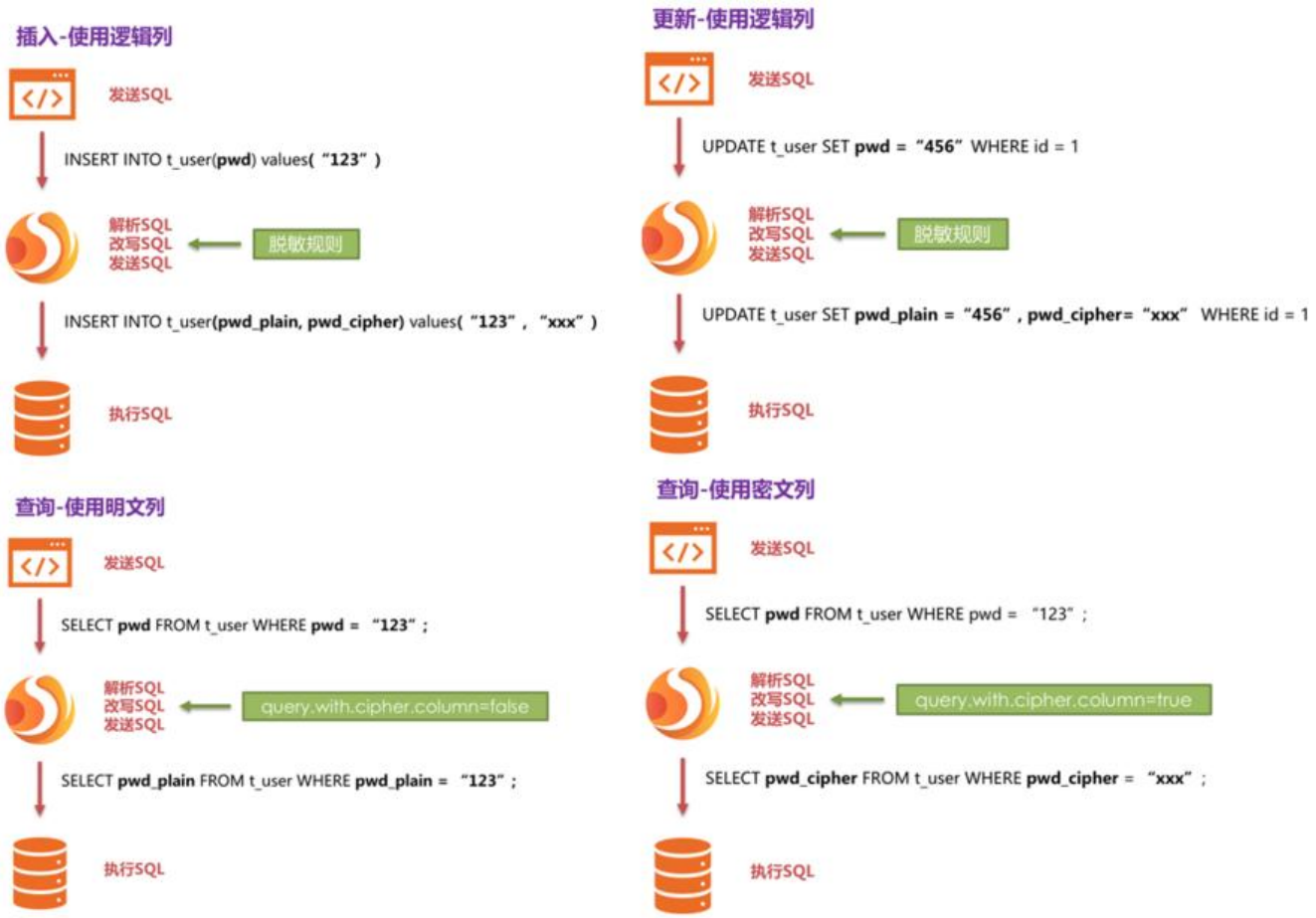



原文链接：Apache ShardingSphere 数据脱敏




原文链接：Apache ShardingSphere 数据脱敏




# 实战体验

## 添加依赖

```
 <!--引入shardingsphere-->
<dependency>
      <groupId>org.apache.shardingsphere</groupId>
     <artifactId>sharding-jdbc-spring-boot-starter</artifactId>
     <version>4.0.0-RC2</version>
</dependency>
<dependency>
    <groupId>org.apache.shardingsphere</groupId>
    <artifactId>sharding-jdbc-spring-namespace</artifactId>
    <version>4.0.0-RC2</version>
</dependency>
```

注意:官方最新文档只适用最新版本，其他版本可能因为兼容性问题无法正常使用。

## 创建数据表

```
CREATE TABLE `t_order` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `pwd_cipher` varchar(2000) DEFAULT NULL COMMENT '密文数据列',
  `pwd_plain` varchar(255) DEFAULT NULL COMMENT '明文数据列',
  PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
```

# 添加ShardingSphere配置信息

注：此处使用yml格式文件，且之前配置的springboot原生的数据源信息要移除，此处可以选择自己欢的数据库连接池。

```
spring:
  shardingsphere:
    datasource: #使用shardingsphere的数据源代替原生数据源
      name: ds
      ds:         #基础数据库信息配置，参数同原生一致
        type: com.alibaba.druid.pool.DruidDataSource
        driver-class-name: com.mysql.jdbc.Driver
        url: jdbc:mysql://127.0.0.1:3306/encrypt?serverTimezone=UTC&useSSL=false
        username: root
        password: root
        max-total: 100
        initialSize: 10
        minIdle: 10
        maxActive: 100
        maxWait: 60000
        timeBetweenEvictionRunsMillis: 60000
        minEvictableIdleTimeMillis: 300000
        validationQuery: SELECT 1 FROM DUAL
        testWhileIdle: true
        testOnBorrow: false
        testOnReturn: false
        poolPreparedStatements: true
        maxPoolPreparedStatementPerConnectionSize: 20
    encrypt:
      encryptors: #定义加密算法
        encryptor_aes: #加密算法1 AES，名字可以自定义
          type: aes
          props:
            aes.key.value: 123456
      tables:       #定义脱敏规则
        t_order:    #目标表
          columns:
            pwd:    #目标表逻辑列
              plainColumn: pwd_plain  #目标表明文列
              cipherColumn: pwd_cipher  #目标表密文列
              encryptor: encryptor_aes     #采用的加密算法
    props:
      sql.show: true                          #打印SQL
      query.with.cipher.comlum: true     #使用密文搜索
```

# 注入数据源

```
import javax.sql.DataSource;

@Resource
private DataSource dataSource;
```

# 适用场景说明

1. 后端数据库为MySQL、Oracle、PostgreSQL、SQLServer。
2. 用户需要对数据库表中某个或多个列进行脱敏(数据加密&解密)。
3. 兼容所有常用SQL

# 限制条件

1. 用户需要自行处理数据库中原始的存量数据、洗数。
2. 使用脱敏功能+分库分表功能，部分特殊SQL不支持，请参考 SQL使用规范。
3. 脱敏字段无法支持比较操作，如：大于小于、ORDER BY、BETWEEN、LIKE等
4. 脱敏字段无法支持计算操作，如：AVG、SUM以及计算表达式

# 更多介绍

参考官方说明文档：
https://shardingsphere.apache.org/document/current/cn/features/orchestration/encrypt/