



Apache Flink

# Flink CDC + Kafka 加速业务实时化

任庆盛 / Apache Flink Committer 2022-9-24

# CONTENT

目录 >>

01 /

Flink CDC 技术

02 /

Flink + Kafka 实时数据集成方案

03 /

Demo

# 01 Flink CDC 技术

# CDC 技术

广义的概念上，能够捕获数据变更的技术，我们都可以称为 CDC (Change Data Capture)。  
通常我们说的 CDC 技术主要面向数据库的变更，是一种用于捕获数据库中数据变更的技术。



**数据同步**  
用于数据备份、系统容灾



**数据分发**  
一个数据源分发给多个下游



**数据采集**  
面向数据仓库/数据湖的 ETL 数据集成

# CDC 实现机制

按照业界主流的实现机制可分为：



## 基于查询的 CDC

- 离线调度查询作业，批处理
- 无法保障数据一致性
- 不保障实时性



## 基于日志的 CDC

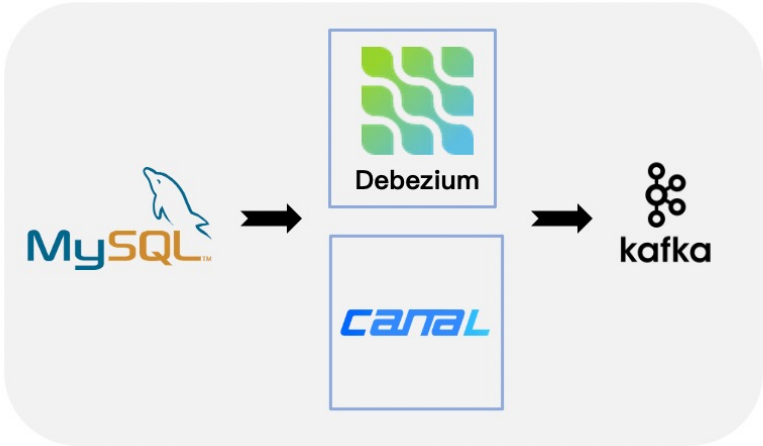
- 实时消费日志，流处理
- 保障数据一致性
- 提供实时数据

# 常见 CDC 方案比较

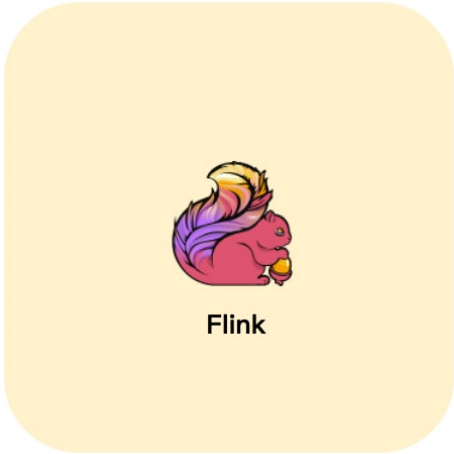
	Flink CDC	Debezium	DataX	Canal	Sqoop	kettle	Oracle Goldengate
CDC 机制	日志	日志	查询	日志	查询	查询	日志
增量同步	✓	✓	✗	✓	✓	✗	✓
断点续传	✓	✓	✗	✓	✗	✗	✓
全量同步	✓	✓	✓	✗	✓	✓	✓
全量+增量	✓	✓	✗	✗	✓	✗	✓
架构	分布式	单机	单机	单机	分布式	分布式	分布式
Transformation	☆☆☆☆☆	☆☆	☆☆	☆☆	☆☆	☆	☆
生态	☆☆☆☆☆	☆☆☆	☆☆☆	☆☆☆	☆☆	☆☆	☆☆☆

# Flink CDC 技术

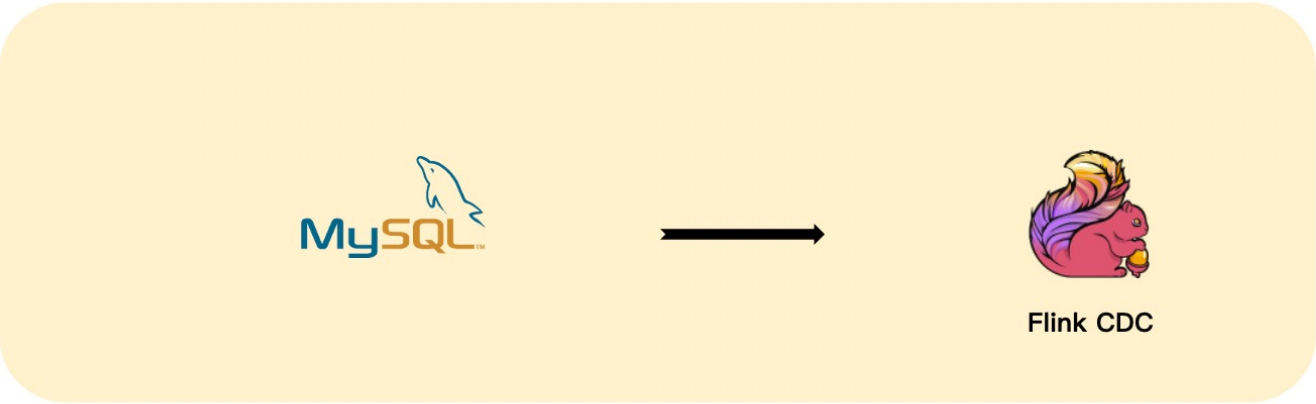
采集 (Extract)



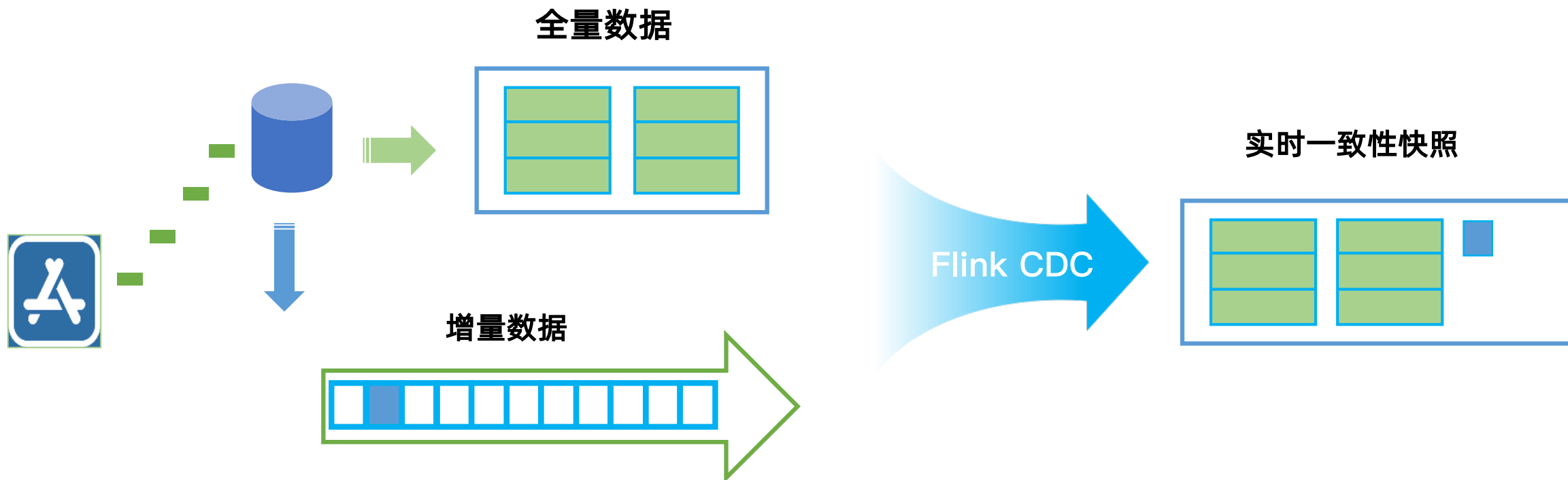
计算 (Transform)



装载 (Load)

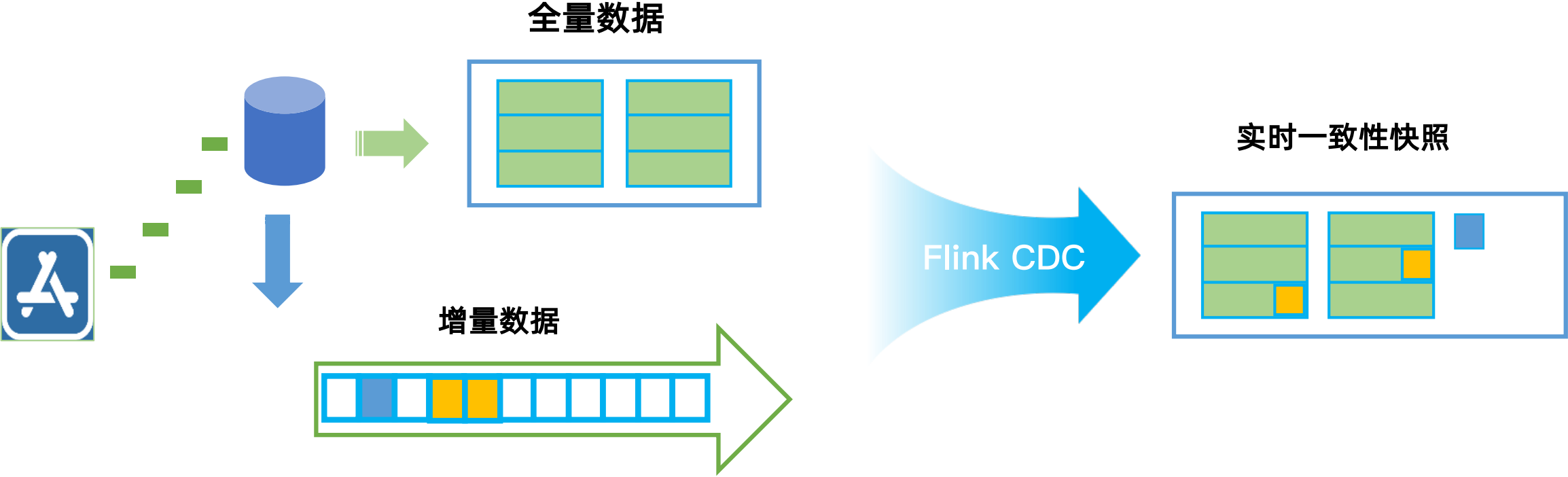


# Flink CDC 全增量一体化框架

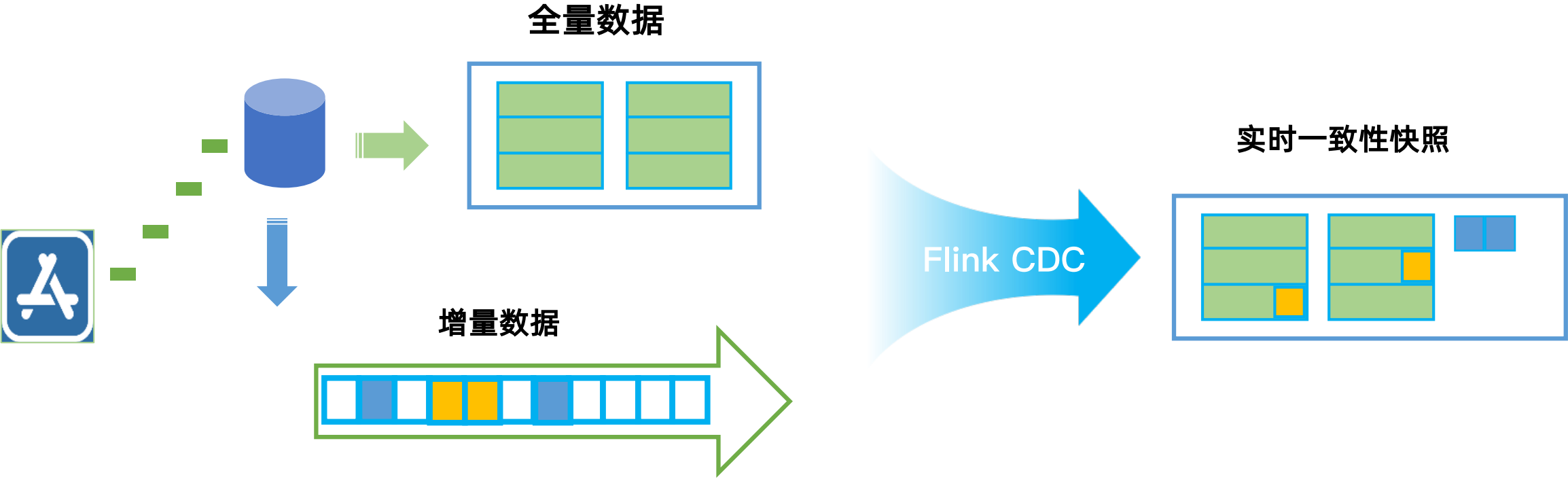




# Flink CDC 全增量一体化框架



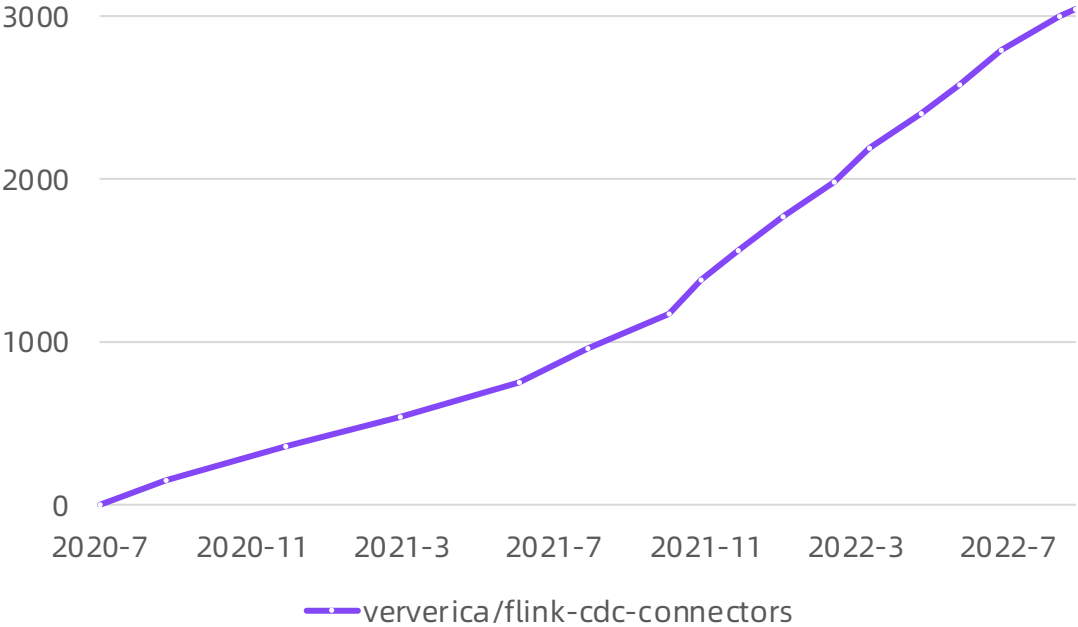
# Flink CDC 全增量一体化框架






# Flink CDC 全增量一体化框架



# Flink CDC 社区



Stars		3079
Commits		879
Forks		1059

Flink CDC  Contributors

社区钉钉群: 33121212

# Flink CDC 2.3 看点

## 稳定性



#1407

MySQL 数据同步加入时区配置检查



#1505

MySQL 解决全量阶段作业重启潜在的数据重复问题



#1523

MySQL 降低 JobManager 内存使用情况

## 新功能



#1363

全面对接 Flink 1.15



#1157

MySQL 支持从指定时间 / 位点启动



#976

Oracle 接入增量快照框架

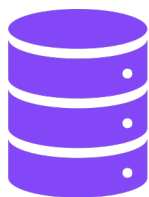
Flink CDC



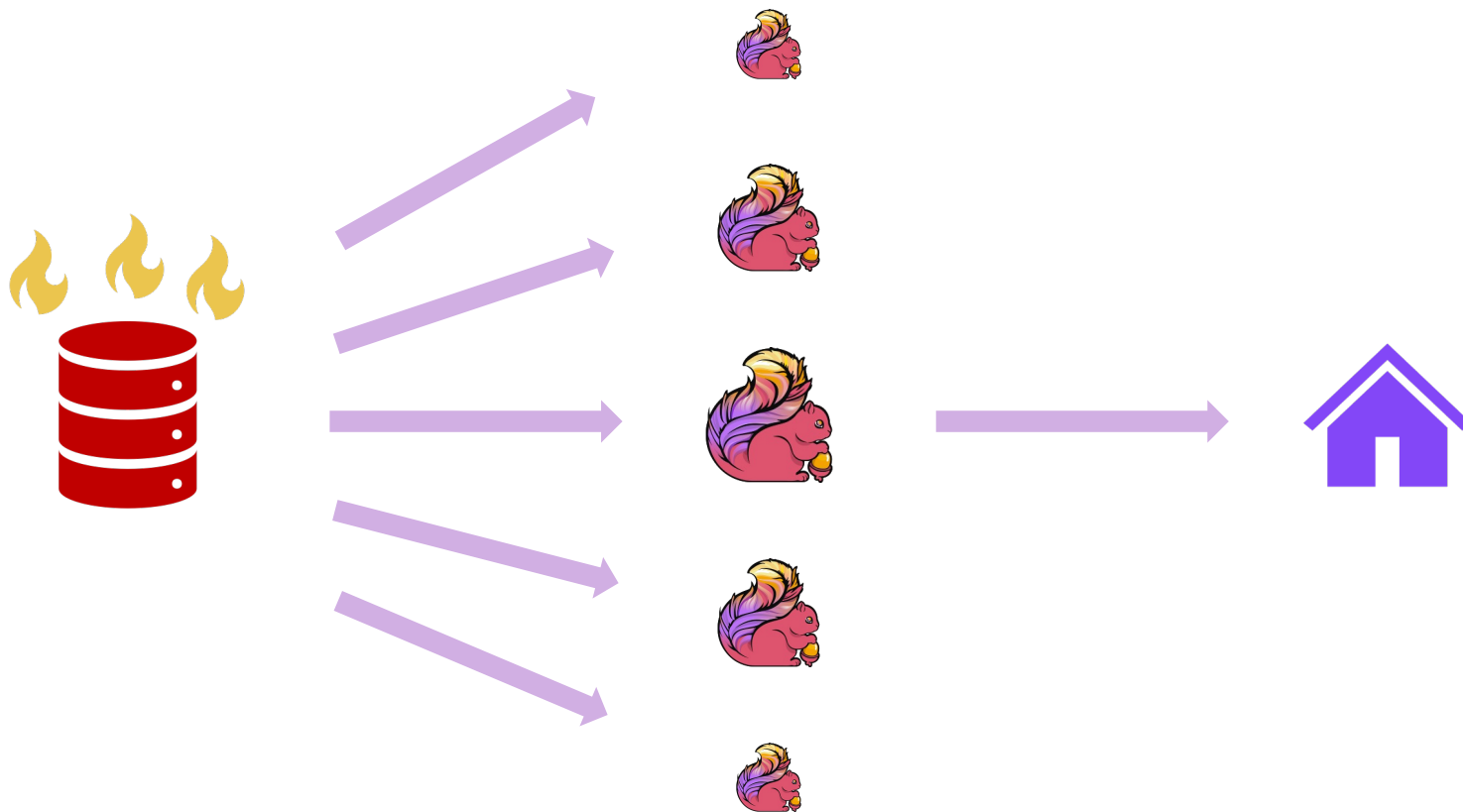
Contributors

## 02 Flink + Kafka 实时数据集成方案

# 业务痛点



## 业务痛点



- 不同作业包含同一个库的 CDC 表
- 过多的同步任务导致数据库负载高
- 热点表 / 数据库频繁访问



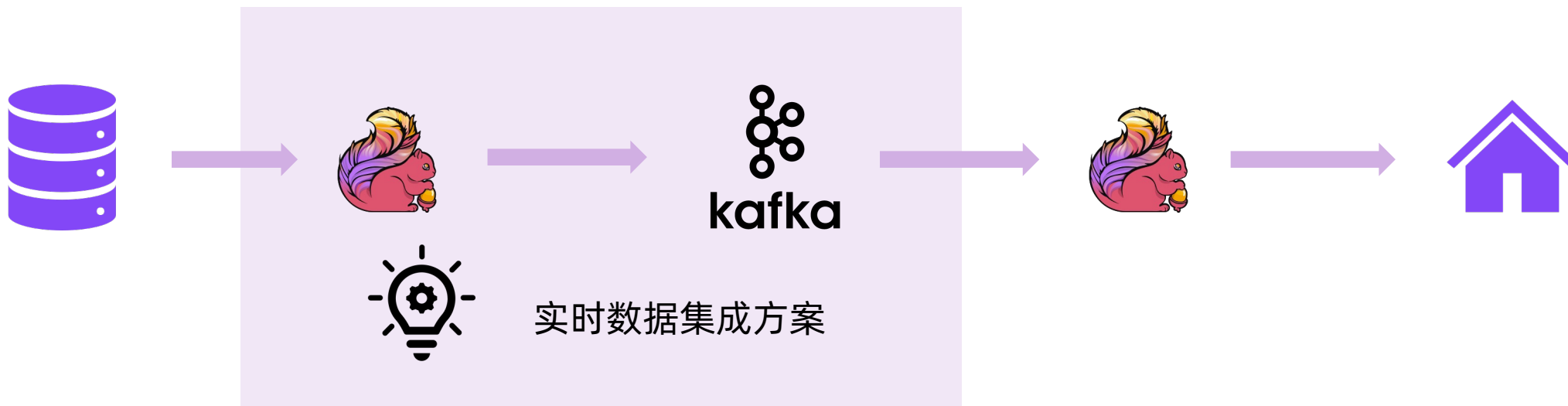
## 业务痛点



- 配置 CDC Source?
- 配置 Kafka Sink?
- 规划 Topic / Partition?

.....

# 实时数据集成方案



## CREATE TABLE AS (CTAS) 语法

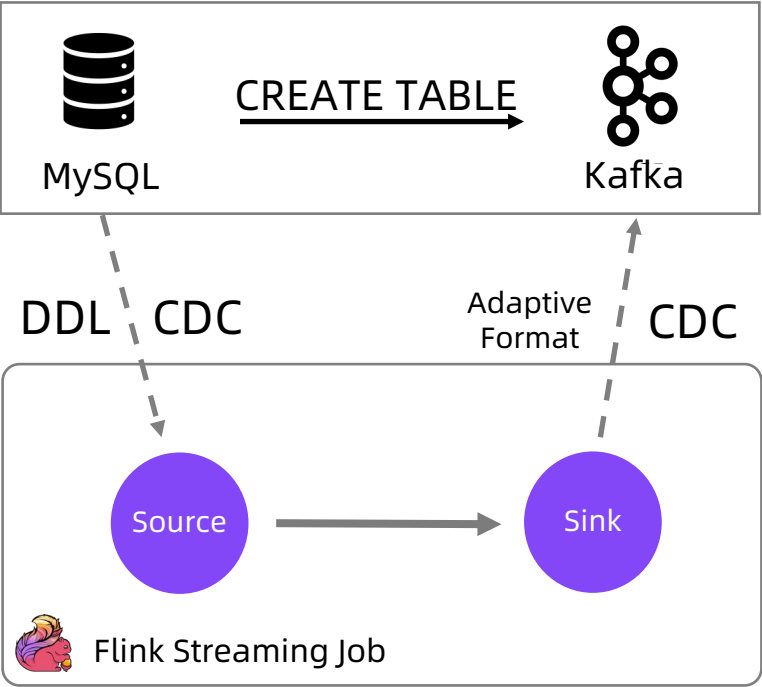
```
CREATE TABLE IF NOT EXIST sink_table  
AS TABLE mysql.mydb.source_table  
/*+ OPTIONS('server-id' = '8001-8004') */
```

## CREATE DATABASE AS (CDAS) 语法

```
CREATE DATABASE IF NOT EXIST sink_database  
AS DATABASE mysql.mydb INCLUDING ALL TABLES  
/*+ OPTIONS('server-id' = '8001-8004') */
```

# 实时数据集成方案

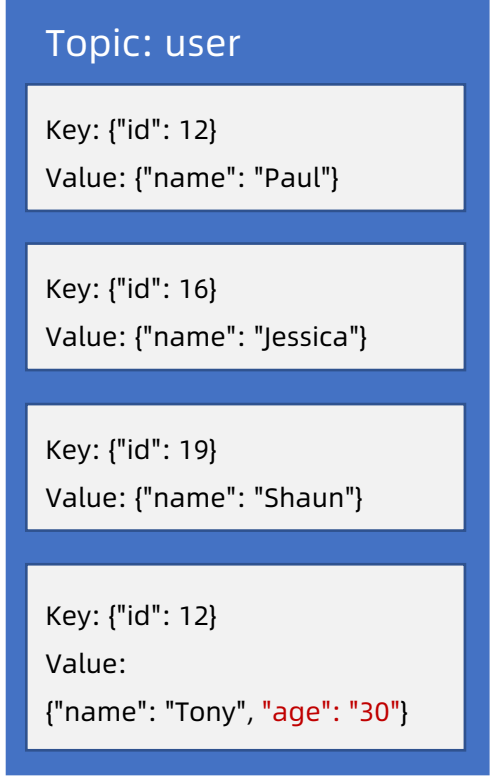
## 表结构变更自动同步



user

id	name	age
12	Paul	NULL
16	Jessica	NULL
19	Shaun	NULL
27	Tony	30

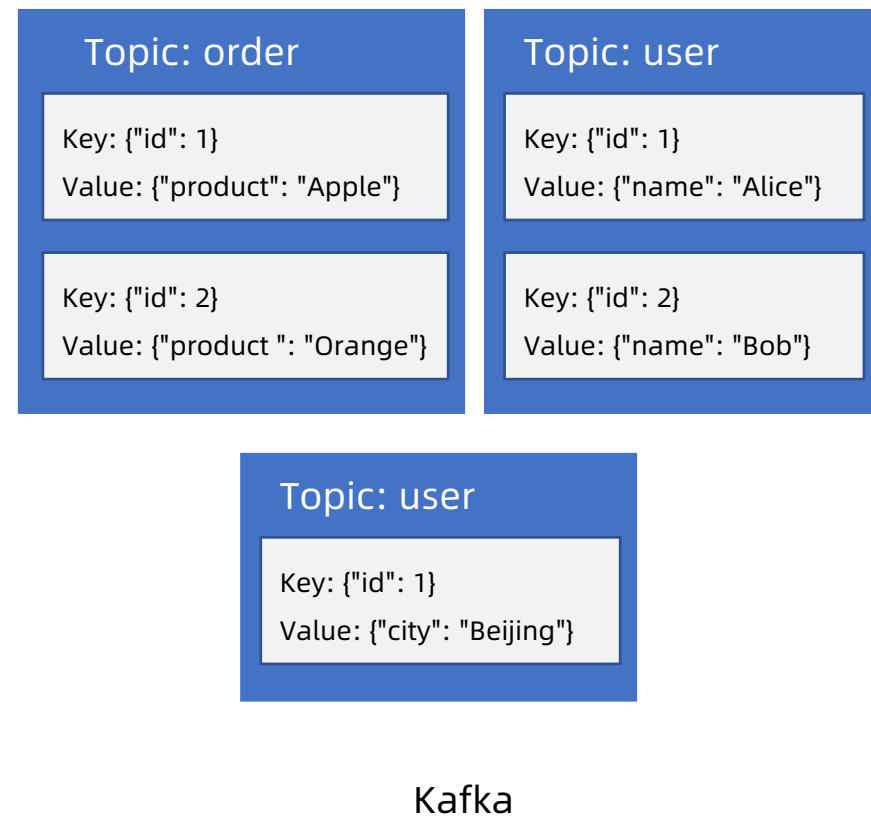
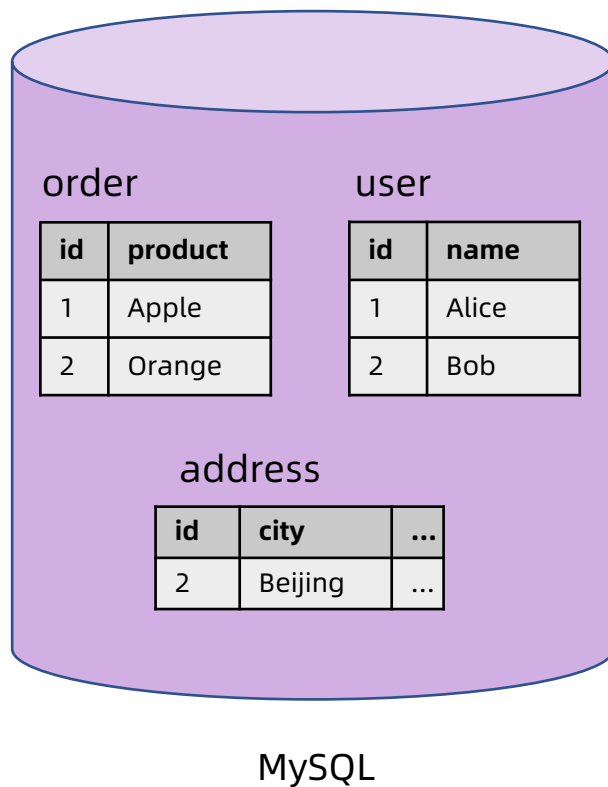
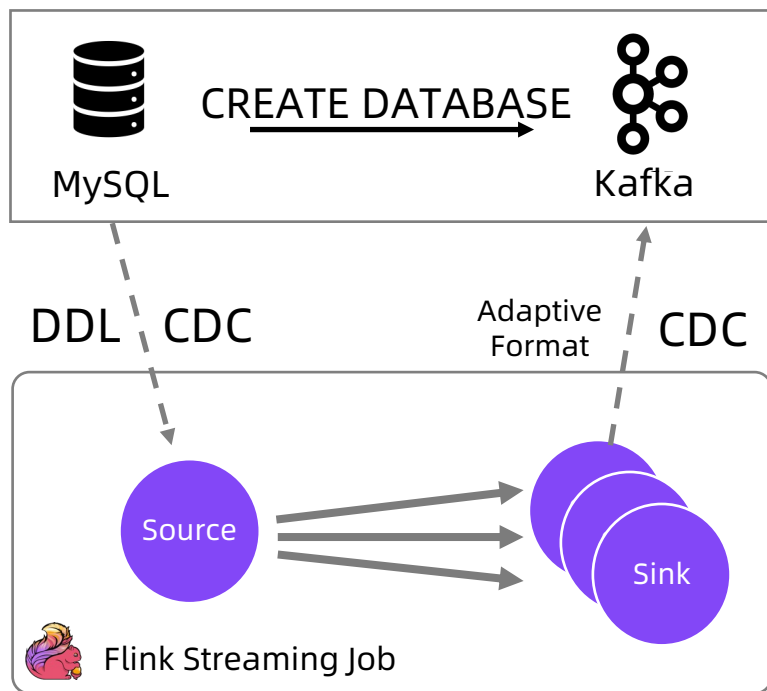
MySQL



Kafka

# 实时数据集成方案

## 整库同步



# 实时数据集成方案

## 01

一条 SQL 语句 (CTAS、CDAS)  
即可同步单表或整库，无需反复配置作业参数

## 02

自动创建 Kafka Topic / Partition  
并建立映射，用户无需在 Kafka 集群进行手动配置

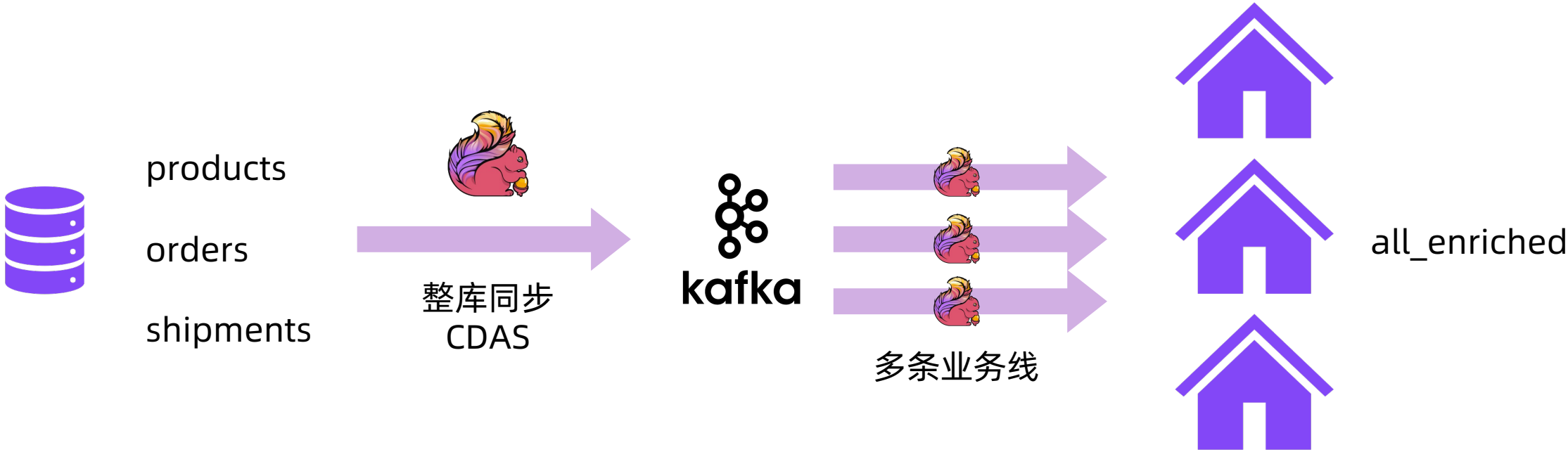
## 03

原生支持添加可空列、删除可空列、重命名列等表结构变更同步策略 (Schema Evolution)

03

Demo: Flink + Kafka 实现 CDC 数据的实时集成和实时分析

Demo



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Flink阿里云实时计算 | 作业开发MySQLDMS - Data Management ServiceKafkaE-MapReduce 管理控制台 | onKafka数据库同步到Kafka测试 - 运维

支持 % 模糊匹配表名称

表可编程对象

all\_enriched

orders

products

shipments

InitChangesEnriched

执行(F8)格式化(F10)执行计划(F9)我的SQLSQL诊断设置任务编排数据可视化跨库查询

1DROP TABLE IF EXISTS products;

2DROP TABLE IF EXISTS orders;

3DROP TABLE IF EXISTS shipments;

4

5-- Products table

6CREATE TABLE products (

7id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

8name VARCHAR(255) NOT NULL,

9description VARCHAR(512)

10);

11ALTER TABLE products AUTO\_INCREMENT = 101;

12INSERT INTO products

13VALUES (default,"scooter","Small 2-wheel scooter"),

14 (default,"car battery","12V car battery"),

15 (default,"12-pack drill bits","12-pack of drill bits with sizes ranging from #40 to #3"),

16 (default,"hammer","12oz carpenter's hammer"),

17 (default,"hammer","14oz carpenter's hammer"),

18 (default,"hammer","16oz carpenter's hammer"),

19 (default,"rocks","box of assorted rocks"),

20 (default,"jacket","water resistant black wind breaker"),

21 (default,"spare tire","24 inch spare tire");

22

23-- Orders table

24CREATE TABLE orders (

25order\_id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

26order\_date DATETIME NOT NULL,

27customer\_name VARCHAR(255) NOT NULL,

28price DECIMAL(10, 5) NOT NULL,

29product\_id INTEGER NOT NULL,

30order\_status BOOLEAN NOT NULL

31) AUTO\_INCREMENT = 10001;

32

33INSERT INTO orders

34VALUES (default, '2020-07-30 10:08:22', 'Alice', 50.50, 102, false),

35 (default, '2020-07-30 10:11:09', 'Sally', 15.00, 105, false),

36 (default, '2020-07-30 12:00:30', 'Edward', 25.25, 106, false);

37

38-- Shipments table

39CREATE TABLE shipments (

40shipment\_id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

41order\_id INTEGER NOT NULL,

42origin VARCHAR(255) NOT NULL,

43destination VARCHAR(255) NOT NULL,

44is\_arrived BOOLEAN NOT NULL

45) AUTO\_INCREMENT = 1001;

46

47INSERT INTO shipments

48VALUES (default,10001,'Beijing','Shanghai',false),

49 (default,10002,'Hangzhou','Shanghai',false),

50 (default,10003,'Shanghai','Hangzhou',false);

51

100共 4 条

1/1

执行历史执行结果1执行结果2执行结果3执行结果4执行结果5执行结果6执行结果7执行结果8执行结果9执行结果10执行结果11执行结果12执行结果13



# Thanks